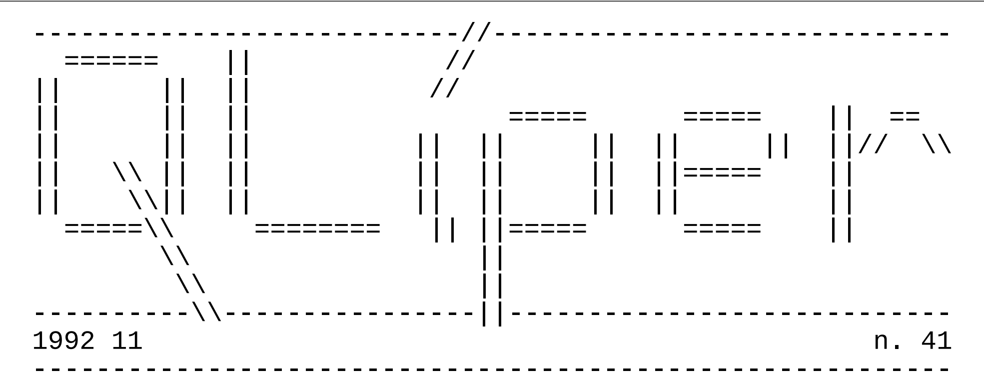


1992 11 n. 41

Contenido Enhavo Contents

QLíper 42, 43...:
-Computación en lengua china sostenida por un banco estructurado de
 conocimiento (Cxinlingva komputado subtenita de strukturita sciobanko)
-Forth: Charles Moore, 1979
-La tercera reunión de QLíper
-PC 486: dónde están los 8 megas, matarilerilerile...!?
-Hermes, el sustituto del 8049
-Más problemas lingüísticos en el mundo del QL (EE.UU., Italia...)
-Conferencia en inglés sobre Clipper con traducción simultánea
 Anglalingva prelego pri Clipper per samtempa tradukado
-Clipper, dBFast, xBase... Aspen

Grupo español de usuarios de QL:	QLíper
Hispana QL-uzantaro:	Acacias 44 (Monteclaro)
Spanish QL Users Group:	E-28223 Pozuelo de Alarcón

Editorial

Esta vez nos llevamos el premio al retraso del año (mea culpa, culpa mía, mia kulpo...!) Sucesos inesperados, cuasiparanormales, como siempre. Y gracias a Miguel Estarellas, que nos ofreció la versión 2 de The Editor, mediante la cual

es más rápido y cómodo componer la revista, no salimos con las uvas, que ya es de agradecer.

Para colmo, muchos no habréis podido leer el QLíper 40 por defecto de la grabación. La unidad de disco me jugó una mala pasada, que se juntó a la casualidad de que (como ya os dije en la circular) no revisé las copias antes de meterlas en los sobres. Por cierto que fue un buen estreno: ¡la primera vez que mandamos QLíper fuera de nuestras fronteras, lo mandamos corrompido! En este disco encontraréis el número anterior, por si acaso no pudisteis leerlo.

En este número 41 encontraréis las habituales y conocidas locuras de nuestro "alma mater" Salvador, unos excelentes artículos de Félix Alonso y Javier Zubieta (quien venciendo su habitual pereza, como él mismo dice, nos había enviado un montón de cosas), más cosas de Pedro Reina, mucho C, pero que mucho mucho por todas partes... ¡La C-manía nos invade! ¿Qué será será lo que tiene el C, además de un nombre así? Creo que debería vacunarme antes de que sea demasiado tarde, porque parece contagioso...

Todas mis colaboraciones, y las traducciones prometidas del inglés y del esperanto se han quedado en promesas, por los mismos motivos por los que llevamos dos semanas de retraso. Os habéis librado por esta vez, pero no escaparéis tan fácilmente...

Acerca de nuestro experimento internacional, os diré que al principio fue decepcionante, pero después empezaron a llegar respuestas. El primero en responder fue Alan Pemberton (*** Thanks, Alan! ***), del grupo escocés, quien ha quedado en mandarnos regularmente su hoja informativa mensual, de la que ya se han recibido tres números. Después, el disco enviado a EE.UU. regresó porque nadie lo recogió, pobrecito. Debe de ser que la dirección aparecida en el SQLW no es correcta. Cambiaremos la dirección en el próximo envío, a la de IQLR. Poco más tarde, llegó el disco 22 de QITALY (*** Grazie, Eros! ***), y los números 28, 29 y 30 de Quasar (*** Danke, Franz! ***). Por cierto, tanto en el boletín del grupo escocés como en el último de Quasar se nos menciona. Ya que nuestro presupuesto es muy ajustado, con este número va a ir una nota a los grupos extranjeros diciendo que este es el último número que recibirán aquellos que no nos hayan enviado sus boletines, es decir, que a partir de ahora el envío de QLíper se hará sólo como intercambio.

Por cierto, veréis que resurge en este número la polémica de los caracteres acentuados y los ordenadores, en una carta de Francisco Díaz-Tendero, algo que habíais planteado ya en alguna otra ocasión, también por teléfono. Me pregunto qué pensarían quienes se toman tan a la ligera el "presunto" problema si los supersignos en castellano no marcasen principalmente las excepciones a la acentuación sino que formasen parte de letras autónomas (como en el caso de la n y la ñ), como en alemán las tres vocales (de las ocho que tiene la lengua) cuya grafía tiene una diéresis. Es más, ¿por qué será que en Quasar escriben nuestro nombre, QLíper, correctamente, pero por más que escribo a QL World no hacen ni caso? ¿Entendéis por dónde van los tiros? Estoy seguro de que también los franceses lo escribirían bien (si nos hubiesen respondido), sin dar por supuesto que el palito sobre la í es una caca de mosca accidental y caprichosa, como incluso muchos castellanoparlantes suponen, por pereza y/o desconocimiento.

Aquellos cuyos discos aún no han retornado, que no piensen que se han perdido.

Hay muchos discos por devolver, y todos irán partiendo por riguroso orden de llegada sin atropellarse. Hasta ahora se han conservado las colaboraciones en los discos originales, y éstos no se han devuelto hasta haber publicado aquéllas, pero como siempre hay que adaptar los textos, recomponerlos, etc., lo que se hará de ahora en adelante es hacer estas operaciones en el momento de recibir los artículos, conservarlos en discos de QLíper hasta para su publicación, y así devolver pronto los discos recibidos, para que no os desesperéis.

Intentad enviar los textos en ficheros ASCII simples. Si no disponéis más que de Quill, mandadlos en un fichero _lis, es decir, imprimiendo en un fichero (pero sin negrita ni subrayado, y quitando antes pies de página, márgenes y todo eso, para que quede lo más limpio posible).

Como veis, en este número no hay indentación o sangría en los párrafos, porque, ya que éstos han de ir separados por una línea en blanco en The Editor, no es

necesario más.

El próximo número, si nada se tuerce, lo compondré a medias con Pedro Reina, para distribuir un poco el trabajo, ya que en Madrid es donde más QL-adictos hay por kilómetro cuadrado, y es más fácil repartirse algunas tareas, evitando así las probabilidades de que vuelva a pasar lo mismo que esta vez.

Bueno, abnegados QL-maníacos, no hay mal que por bien no venga, y si este número es el más tardío de nuestra modesta pero gloriosa historia, también es de los más rollizos, así que podéis disponeros a hincarle el diente con calma, porque tenéis para rato. Y ya os dejo, pues se os nota que estáis ávidos de empezar a leer. Hasta dentro de seis semanitas (¡prometido!).

Marcos, 1992 11 14 (!)

PD: el asunto de la cuota del año 1993 está detallado aparte, buscadlo al final del índice.

El buzón

Bilbao, 12 de Octubre de 1992.

Estimado amigo Marcos,

De manera súbita e inesperada me he puesto delante del teclado y he escrito un rollo plúmbeo de los que me caracterizan, que espero pueda ser incluido en próximos números de QLíper. Si no resultan útiles e informativos, al menos servirán como somnífero en noches de insomnio.

La verdad es que me he sentido inspirado y, aprovechando el largo puente del Pilar, me he decidido a escribir un largo texto, dividido en varias partes. La primera es una oferta, en la que que pongo a la venta mi QL. No te preocupes, eso no significa que vaya a abandonar el mundo del QDOS. En la misma oferta explico mis motivos.

La segunda parte se refiere a una pequeña reflexión acerca de aspectos no muy conocidos de los PCs y que nos puede servir de consuelo a nosotros, los parias de la tierra, que no nos hemos doblegado aún a los Pcs.

La tercera parte trata de algunas novedades en el mundo QL-Atari, con referencia al folleto de septiembre de Jochen Merz.

Por último, hago mención a la última novedad en el mundo Atari: el nuevo ordenador Atari Falcon. No es que tenga mucha relación con el mundo QL, pero puede que resulte interesante para algunos.

La verdad es que me he extendido más de lo habitual. Si quieres, puedes cortar mi aportación en cuantas partes estimes oportuno (por cierto, el último número de QLíper, el 40, quedó un poco pobre, con poco material. A ver si nos animamos todos a colaborar o, de lo contrario, se acabó lo que se daba).

Te mando un archivo en formato ASCII, que escribí originalmente en mi Atari Mega ST2, y que luego pasé a formato QL mediante una utilidad incluida con la tarjeta QL-Emulator. Lo hice para probar y me ha salido bastante bien, aunque hay algo que me revienta, y es como tratan los distintos programas los saltos de línea y la longitud de línea. Verás que hay algunas líneas que son más largas de lo que debían. No he sido capaz de arreglar esto con el editor de Metacomco, y después de tanto esfuerzo escribiendo, la vagancia intrínseca que me caracteriza ha dejado esto sin solución. Seguro que tú lo solucionas sin ningún problema. Perdona los inconvenientes que te pueda causar esto.

Sin otro particular me despido. Saludos,

Javier Zubieta

DE: Francisco Díaz-Tendero de la Flor
FECHA: 1992_07_29

SOBRE EL CORREO

--Gracias por reenviarme el disco a mi dirección particular. Ya me estaba pareciendo que tardaba, y esta es la explicación.

--Creo que esta ocurriendo algo raro con el correo, y he comenzado por investigar en la oficina de la base; ya que lo lógico es que el problema estuviese aquí, al ser un sitio en el que trabajan no profesionales, (soldaditos y marineros de reemplazo).

--Me ha atendido muy bien el jefe de la estafeta, (que sí es un funcionario del Cuerpo de Correos), y me ha demostrado que el sobre no llegó aquí en ningun momento, porque no tiene el sello de esta oficina, que ellos siempre le ponen con la fecha de reexpedición; como creo que es obligatorio segun sus reglamentos.

--Además el sello de devolución de esta oficina es distinto. Así que no cabe la duda, lo que quiera que halla pasado no ha sido allí.

--A decir verdad el sobre no muestra ningún sello legible que permita trazar su recorrido, y saber desde dónde se devolvió. Esto, unido a que le faltan los sellos originales hace pensar al jefe de aquí que a "alguien" pudieron antojárseles, y lo devolvió desde cualquier sitio. Naturalmente suponiendo que no se los hayas quitado tu mismo [1] antes del segundo envio. Si yo le hubiera podido asegurar esto , según el, "la cosa podría traer cola". A mí se me ocurre pensar que ya que robaron los sellos, si es que fue asi, también podrían haberse quedado con el disco entero.

--Aunque nada de esto haya ocurrido, y sólo se trate de un despiste de algun funcionario aburrido, queda el caso de que tanto este disco, como el otro que te mando, me han llegado con muchas corrupciones. Y esto no me pasa normalmente con los que pido a Inglaterra; no creo que los circuitos nacionales sean más estrictos en cuanto a inspecciones con aparatos que puedan provocarlas.

--Por esto creo mejor que los sigas mandando al trabajo, como antes; y si puedes trates de ver si ha podido pasar algo en la oficina desde la que normalmente los remites. Si vuelve a haber problemas habría que iniciar una reclamación en forma; pero esto tiene que hacerlo el remitente.

--Yo lamento el exceso de molestias , y coste, que esto significa; pero no es cuestion de estar girando cantidades tan pequeñas. Si te parece bien hacemos cuentas y te mando el exceso con la cuota del año que viene.

SOBRE EL CLUB

--Os debo una colaboración que me pidió Salvador Merino sobre los discos duros. No se me ha olvidado. El problema es que mi mala salud no me ha dejado tiempo prácticamente en todo lo que va de año. No tengo nada grave, pero una rinitis crónica hace que me duela la cabeza casi todos los días durante algunas temporadas, y así no hay manera. Si puedo os mandaré algo en este mismo disco.

SOBRE LA REVISTA

--Aunque te devuelvo el disco para ver si puedes regrabármelo, he podido leer alguna cosa, y te quisiera comentar; como ya creo que hice en otra ocasión, que estoy completamente de acuerdo en pasar del Quill y sus _doc.

--Hace ya tiempo que no lo empleo para nada, y el primer problema que se me presenta cuando recibo vuestro disco es el transformar todos los ficheros a texto llano, (con la utilidad que acompaña al Editor). Esto es un trabajo extra, pero la única manera de no depender de un progrma concreto; bastante malo y caprichoso por cierto.

--Los ficheros de texto llano no sólo tienen las ventajas que tú dices, sino que ademas:

--Son completamente transportables, incluso para los que padecen MS-DOS en su trabajo, y se los quieren llevar con el XOVER.

--Son más cortos, y ahora se pueden manejar con el editor que viene con el C68

que es una versión mejorada del antiguo Metacomco. Creo que esta es la mejor solución para la gente que esté corta de memoria, y no pueda cargar el Editor.

--En cambio no me parece una buena idea el poner toda la revista en un sólo fichero. Esto la hace mas difícil de manejar y no tiene ninguna ventaja a la hora de imprimir.

--Lo bueno de una revista en disco es que se guarda mucho mejor que una clásica de papel, y en caso de desearlo, lo normal es que no se imprima toda sistemáticamente, sino sólo las partes que se necesitan en cada momento.

--Por esto lo que yo sugeriría sería una división en capítulos; más o menos como ahora, con un fichero índice al principio, que podría contener un párrafo de resumen orientativo sobre cada uno de ellos. Esto facilita hacerse una idea global al principio, y después ir derecho a lo que se quiere leer en cada momento.

--Si se desea imprimir toda la revista de un tirón, tampoco es problema, con cualquier copiador automatico, o semi; la misma orden WCOPY [2].

--El editor que te digo es de dominio publico, así que no hay problema legal en pasárselo a alguno que todavía no lo tenga.

--Otra cosa que habrás observado es que paso de muchas normas de

ortografía, que en un ambiente informático no hacen más que complicar la vida con caracteres difícilmente tratables, sin mejorar la legibilidad del texto. Al fin y al cabo si el inglés se puede escribir sin acentos, creo que el castellano también [3].

--Esto es especialmente importante en los nombres de fichero, por poner una letra acentuada, ya te ves obligado a tratarlos como cadenas, y es una molestia sin compensación. Lo sugiero tambien para el nuevo formato de la revista.

--En fin, que lo pases lo mejor posible en las vacaciones, y espero poder colaborar un poco más en el futuro.

Recuerdos.

Red.:

Estimado Francisco: en primer lugar te agradezco mucho que nos hayas enviado material, y espero que te mejores y nos puedas seguir mandando más. Recibí tus discos en mal estado, y te los devuelvo regrabados (uno estaba muy mal y lo he sustituido por otro) por correo certificado en sobre aparte del QLíper 41. Perdona el retraso. Por cierto, he encontrado un sitio en el que me pueden encargar las hojas para archivar discos, como las que estabas buscando. Te informaré en el próximo número (la que me mandaste, hecha artesanalmente, te la devuelvo; es una muy buena idea).

Quisiera hacer algunas observaciones a lo que dices en tu carta:

[1] No recuerdo si quité los sellos o no. Suelo conservar los sellos usados, porque tienen valor para mucha gente (parroquias, asociaciones benéficas, etc.) Creo que es para venderlos para crear colecciones de sellos usados. Además, soy una panacea de proveedor, porque me llegan continuamente cartas y envíos de muchos países de todo el mundo. Pero hace mucho tiempo de eso, y no sé decirte si me quedé con los de tu carta devuelta o no. Veremos qué pasa la próxima vez; te vuelvo a enviar todo a la base.

[2] Creo que WCOPY no sirve para copiar a la impresora, porque trabaja con comodines en los nombres. No sé si hay algún truco para hacerlo.

[3] Con todo respeto, nunca he oído tantos absurdos en tan pocas sílabas. Tu razonamiento y argumentos, si es que se pueden llamar así, parten de graves errores de principio. Me pillas con poco tiempo (aún quedan muchísimas tildes que poner en este QLíper antes de poder enviarlo), así que te contestaré de forma un tanto desordenada:

Según tú pues, si el búlgaro se escribe en cirílico, ¿por qué no el swahili?; si el chino se escribe en Hanzi, ¿por qué no el serbocroata? Al fin y al cabo pues,

si los ordenadores los hubiesen inventado los alemanes y el código de caracteres más extendido estuviese basado en su alfabeto (cosa que hubiera podido pasar perfectamente), te parecería "natural" usar los siguientes símbolos: ö ä ë ß, porque su posición en el juego de caracteres de todos los ordenadores estaría normalizada, cosa que ahora no es así. ¿Qué pensarías entonces? El problema no es "supersignos sí o supersignos no". La diferencia entre lo que te parece "natural" y lo que te parece incómodo pues, no está en la lengua en sí sino en la costumbre obligada por ciertas características de la máquina, producto del azar, y susceptibles de ser alteradas. Debe ser la técnica la que se adapte al ser humano (y a sus lenguas), y no al revés.

No obstante, si la mayoría de suscriptores pide que no sea corregida la ortografía (aparte de alguna otra cosa que surge de vez en cuando, como preposiciones mal usadas, formas verbales cruzadas, géneros y números incorrectos, anglicismos...) de QLíper, yo democrática y gentilmente cedo mi puesto de inmediato al primer voluntario, porque quiero leer una revista en castellano, no en "espanglish". Por cierto, espero que no te ofendas porque haya sembrado tus cartas con incómodas tildes, y le haya añadido a tu apellido la sílaba que le faltaba, porque el símbolo "i" en castellano representa dos sonidos diferentes, por si no te habías "coscao" de ello todavía, y en ocasiones hay que usar una tilde no para indicar una excepción a la norma de acentuación, sino para cambiar su valor fonético en dicho caso, cuando lo más fácil, sencillo, eficaz, lógico, y por tanto natural (la ley de la máxima eficacia con el mínimo esfuerzo), sería que ambos sonidos se representasen con dos símbolos diferentes, ¿no te parece? Pero ya ves, curiosidades de la madre naturaleza... mejor dicho, caprichos o errores de quienes (pobres seres humanos como tú y como yo) se han ido inventando, con lápiz y papel delante (o con pluma de ganso o con boli, según la época), las reglas del idioma que usamos, esas mismas reglas que ahora nos parecen tan "naturales", (¡como si hubieran nacido de debajo de un repollo o las hubiera traído de París una cigüeña, no te jode!)

Pero entre líneas de lo que dices creo leer una queja que tiene algo de razón: hay sistemas mucho más eficaces para mejorar la comprensibilidad pasiva y la expresión activa, facilitar el aprendizaje y eliminar los errores más comunes a nivel ortográfico, en suma, hacer más eficaz el proceso de la comunicación. Pero sería necesaria una reforma radical de la ortografía actual (si el castellano tiene una ortografía fonética, como muchos inocentes creen, que me parta un rayo ahora mismo, o mejor que se funda el transformador mi QL, que es mucho peor). Pero entonces más de uno de esos ilustres, cultos y apolillados académicos con glorioso lema de limpiametales nos haría crucificar o empalar (según el gotoso pie con el que se hayan levantado esa mañana) después de mearse de risa, igual que los científicos de hace un siglo se rieron a lágrima viva en la cara de Pasteur, ¡ante la absurda y ridícula idea de que algo tan pequeño como un microbio pudiera matar una vaca! Para cualquiera de nosotros hoy, es natural y comprensible que un microbio puede matar no ya una vaca, sino un montonazo de vacas una tras otra, y podemos probarlo porque tenemos conocimientos objetivos sobre la materia, pero entonces no sabían nada de nada, y por pereza intelectual se limitaron a rechazar la idea como imposible sin más preámbulos, al contrario que el abnegado Pasteur (que se dedicó a investigar y a contrastar sin prejuicios ni tabúes, para encontrar la verdad pasando si fuera necesario por encima de los convencionalismos intelectuales de su época). Los ejemplos así se agolpan en la Historia, uno tras otro, en la ciencia, en el arte... Puede que dentro de cien años, los libros (perdón, los vídeos interactivos) hablen de las inútiles y heróicas tildes de QLíper, al lado de una bonita foto de la vaca de Pasteur. En fin, así es la "intelectualidad" de todas las épocas, Francisco, pero a pesar de ellos vamos avanzando, Galileo tras Galileo, poco a poco. Y basta de charla, o más de uno me va a "dimitir". ¿Habéis visto con qué elegancia y con qué morro se puede pasar de hablar de tildes a hablar de ganado vacuno? Todo está relacionado, en este pequeño, primitivo y moribundo planeta.

DE : Francisco Díaz-Tendero de la Flor
FECHA: 1992 Sep 15

--Querido colega. Te adjunto esta carta, con la que tenía escrita y pendiente de enviar desde este verano, como puedes ver por la fecha.

--El problema de las corrupciones sigue como antes o peor. En este último disco ni se podía leer el directorio, y ademas hacía, y hace ruido mecánico. Te envío este, y otros dos anteriores para ver si consigues regrabármelos, y te sugiero

que observes si a otros les pasa lo mismo, sobre todo si los discos pasan por las mismas oficinas.

--Se puede sospechar que "alguien" esté intentando leerlos por el camino con un PC, que es toda la informática que conoce la mayoría; y esto cause las corrupciones. Aparte de ser un delito de violación de correspondencia... Por esta razón habría que hablar con Correos, claro que tendrías que ser tú como remitente. Yo no puedo.

--La razón de la tardanza en contestar, ha sido (aparte de mi dolor de cabeza crónico, el que quería mandar un comentario para la revista sobre los discos duros, como me pidió Salvador Merino, y no he podido.

--Por lo que se refiere al nuevo formato de la revista coincidís casi completamente con lo que yo habría propuesto, sólo que me parece mejor descomponer un gran fichero en partes, para su mas fácil manejo [1]. Total siempre se pueden juntar con The Editor, o cualquier filtro CCT o CAT de los varios que hay o se pueden hacer fácilmente.

--En fin gracias, perdona las molestias , y a ver si me operan esta maldita nariz, y me duele la cabeza un poco menos, para poder dedicar más atención a muchas cosas, entre ellas a vosotros.

Red.:
[1] Yo creo que es más fácil manipular un único fichero. Desde luego, sí para hacer la revista. Para los suscriptores suponía que también, ya que cada cual puede (más sencillo que unirla) imprimirla toda con un solo "COPY", por lo que no hace falta tener ampliación memoria, o cargarla en un editor e imprimir sólo las partes que quiera. El antiguo "follón" de tener varios ficheros no os lo podéis imaginar, a la hora de montar la revista. En cualquier caso, quisiera oír más opiniones, para que entre todos vayamos haciendo los cambios que consideremos beneficiosos.

"Concept": ¡Un compatible QL desde Rusia!

"Concept": QL-Kongrua el Rusujo!

"Concept" (Concepto): Un compatible QL, potente y rápido; una máquina basada en un MC68040, desde Rusia.

Mensaje número 1 sobre Concepto, fechado el 18 de septiembre de 1992.

Andrew Lavrov y Anatoly Tishin, ambos rusos, proyectan diseñar y construir un clónico muy potente del Sinclair QL tan pronto como sea posible. La máquina que tienen en mente no sólo sería capaz de actuar como un QL, ¡sino también como otros ordenadores basados en procesadores MC680xx (Atari ST, Commodore Amiga, Apple Macintosh, etc.) sin ningún cambio en el járgüer! Las características principales de la máquina todavía no son definitivas, y están abiertas a cambios, pero de momento se prevén las siguientes:

-Nombre del proyecto: "Concepto".

-No se trata una mejora de una máquina que ya existe, sino que es un ordenador personal nuevo.

-Procesador Motorola MC68040.

-Frecuencia de reloj sobre los 40 MHz.

"Concept" (Koncepto): potencan kaj rapida QL-kongrua; MC68040-bazita masxino el Rusujo.

Mesagxo numero 1 pri Koncepto, datita 1992e septembre 18an.

Andrew Lavrov kaj Anatoly Tishin, ambaux rusaj, projektas dezajni kaj konstrui potencan klonon de Sinclair QL, lauxeble plej baldaux. La masxino kiun ili celas agivus ne nur kiel QL, sed ankaux kiel aliaj MC680xx-bazitaj komputiloj (Atari ST, Commodore amiga, Apple Macintosh, ktp) sen ajna hardvo-sxangxo! La precipaj ecoj de la masxino ankoraux ne definitivas, kaj estas malfermaj por sxangxigxoj, sed nuntempe antauxvideblas jenaj:

-Projektonomo: "Koncepto".

-Ne temas pri plibonigo de jama masxino, sed gxi estas nova persona komputilo.

-Procezilo Motorola MC68040.

-Horlogxa frekvenco cxirkaux 40 MHz.

-Zonas de RAM y ROM movibles por medio de redireccionamiento programable, para la compatibilidad con los ordenadores existentes basados en un MC680xx.

-Periféricos flexibles de programar (teclado, pantalla, sonido, disqueteras, etc.), basados en micrordenadores de un solo chip.

-Direccionamiento programable de los periféricos.

-Soporte para algunos sistemas operativos estándar, como Unix y OS/9.

-El proceso de arranque permitirá reconfigurar el sistema y seguir trabajando en diversos modos de emulación (QL, ST, Amiga, Mac...) El número exacto de ordenadores que serán emulados dependerá de la popularidad de cada sistema, y del interés de los expertos en unirse al proyecto. El primer ordenador que será emulado es el Sinclair QL.

-No supondrá violación alguna de los derechos de copia del sófgüer en ROM que ya existe.

-Será construido en Rusia con chips importados.

-Precio de mercado estimado: 1.500 dólares estadounidenses.

El MC68040 es un auténtico procesador de 32 bits (con bus de datos de 32 bits) y tiene una memoria caché interna muy rápida. Se espera que el sistema arriba descrito funcione al menos 6 veces más rápido que un QL con Gold Card (es decir, más de 25 veces la velocidad de un QL normal). ¡Esto supone, por ejemplo, que el programa emulador "PC Conqueror" sería tan rápido como una máquina 80386 a 16 MHz, y que los emuladores de Spectrum serían al menos el doble de rápidos que un Spectrum de verdad!

Las personas actualmente implicadas en el proyecto son:

1) Programación del sófgüer del sistema: Andrew Lavrov, Tver, Rusia.

2) Desarrollo del járgüer principal: Anatoly Tishin, Dubna, Rusia.

3) Sófgüer y coordinación del grupo: Carlo Delhez, Steenbergen, Holanda.

No obstante, el proyecto es bastante complicado, así que queremos cooperar con tantos usuarios y expertos como sea posible, de los diferentes sistemas 680xx, en orden a alcanzar

-Moveblaj RAM- kaj ROM-zonoj per programebla aliadresigo, por kongrueco kun jamaj MC680xx-bazitaj komputiloj.

-Elaste programeblaj periferieroj (klavaro, ekrano, sono, diskiloj ktp), bazitaj sur unublataj mikrokomputiloj.

-Programebla readresigo por periferieroj.

-Subteno por iuj normaj operaciaj sistemoj, kiel Unix kaj OS/9.

-La startproceso ebligos readapti la sistemon kaj plu labori laux diversaj emulmodaloj (QL, ST, Amiga, Mac...) La ekzakta nombro da emulotaj komputiloj dependas de la populareco de cxiu sistemo, kaj de la intereso aligxi al al la projekto far spertuloj. La unue emulota komputilo estas Sinclair QL.

-Gxi ne kontrauxos kopirajtojn de jamaj ROM-softvoj.

-Oni gxin konstruos en Rusujo per importotaj blatoj.

-Taksata vendoprezo: 1.500 usonaj dolaroj.

MC68040 estas vere 32-bita procezilo (kun 32-bita datumbuso) kaj havas tre rapidan kacxeon. Oni esperas, ke la supre priskribita sistemo funkcios almenaux 6-oble pli rapide ol QL kun Gold Card (t.e. 25-oble pli rapide ol normala QL). Tio implicas, ekzemple, ke la softva emulilo "PC Conqueror" estus same rapida kiel 80386-masxino je 16 MHz, kaj ke la Spectrum-emuliloj estus almenaux duoble rapidaj ol vera Spectrum!

Jen la homoj nun okupigxantaj pri la projekto:

1) Sistemsoftva programado: Andrew Lavrov, Tver, Rusujo.

2) Cxefhardvara developado: Anatoly Tishin, Dubna, Rusujo.

3) Softvaro kaj grupkunordigado: Carlo Delhez, Steenbergen, Nederlando.

Tamen, la projekto estas suficxe komplika, do ni volas kunlabori kun lauxeble plej da uzantoj kaj spertuloj de la diversaj 680xx-sistemoj, cele atingi bonajn rezultojn. Ankaux de

unos buenos resultados. También a aquellas compañías que deseen cooperar, o contribuir al desarrollo de este potente e innovador nuevo sistema, se les pide que respondan.

firmaoj dezirantaj kunlabori aux kuntribui al la developado de cxi tiu potenca, sxangxa kaj nova sistemo, ni petas respondon.

Se va a organizar una asociación (llamada "Asociación Concepto"), que estará abierta para todos los usuarios, expertos y compañías interesados en el proyecto, y que deseen apoyarlo y tomar parte en el diseño de Concepto.

Organizotas asocio (nome "Koncepto-Asocio"), kiu restos malferma al cxiuj uzantoj, spertuloj kaj firmaoj interesitaj pri la projekto, kaj emaj subteni gxin kaj partopreni la Koncepto-dezajnon.

Si todo esto es de tu interés, haz lo siguiente:

Se cxio cxi interesas vin, agu jene:

Expertos en ordenadores:

Komputilaj spertuloj:

Si estáis dispuestos a colaborar o a contribuir al proyecto, enviad una carta describiendo vuestras aptitudes (conocimientos de járgüer, experiencia en sófgüer, dominio de sistemas concretos basados en un 680xx, etc.) y cuánto tiempo podríais dedicar a este proyecto (como voluntarios). ¡Si tenéis ya algunas ideas que pudieran ser útiles, mencionadlas! Responderemos tan pronto como se haya logrado reunir un grupo de expertos suficientemente grande en los diferentes campos, y cuando sus sugerencias hayan sido evaluadas por Andrew y Anatoly (¡los cerebros del proyecto!)

Se vi pretas kunlabori aux kontribui al la projekto, sendu leteron priskribantan viajn kvalifikojn (hardvarkonoj, softvarsperto, regado de difinitaj 680xx-bazitaj sistemoj, ktp) kaj kiom da tempo vi povus dedicxi al cxi tiu projekto (volontule). Se vi jam havas kelkajn ideojn eble utilajn, menciu ilin! Ni respondos post ol kunigi suficxe grandan grupon de spertoj pri la diversaj fakoj, kaj kiam iliaj sugestoj taksitos far Andrew kaj Anatoly (la projektocerboj!)

Usuarios de ordenadores:

Komputil-uzantoj:

Por favor, mostrad vuestro interés en este proyecto enviando una carta personal a la dirección de abajo. ¡Necesitamos saber realmente cuánta gente está en principio interesada en este sistema! Si además incluís algunos vales de respuesta internacional, os mantendremos informados del tema por correo.

Bonvolu esprimi vian intereson pri cxi tiu projekto sendante personan leteron al la suba adreso. Ni bezonas scii, kiom da homoj apriore interesigxas pri cxi tiu sistemo! Se krome vi aldonos kelkajn irkojn, ni posxte informados vin pri la afero.

Compañías:

Firmaoj:

Cualquier tipo de interés por parte de las compañías es cordialmente bienvenido. El patrocinio y el apoyo para la futura fabricación y distribución serán esenciales.

Ajna intereso far firmaoj estas kore bonvena. Patronado kaj subteno por ontaj fabrikado kaj distribuado estas nepraj.

Todos:

Cxiuj:

¿Pensáis que este ordenador es factible? Creemos que podría competir con los sistemas actuales basados en un 680xx, gracias a su flexible diseño. ¿Cuál es vuestra opinión?

Cxu vi opinias cxi tiun komputilon farebla? Ni kredas, ke gxi povus graux sia elasta dezajno, konkurenci kun nunaj sistemoj 680xx-bazitaj. Kio estas via opinio?

Dirección postal del proyecto:

Posxtadreso de l' projekto:

Carlo Delhez
Emmastraat 3
NL-4651 BV Steenbeergen

Carlos Delhez
Emmastraat 3
NL-4651 BV Steenbeergen

Holanda

Te pedimos por favor que pases este mensaje a tantos usuarios, grupos, revistas y BBSs (de QL, ST, Amiga y Mac) como sea posible. Cuanta más gente alcancemos con esta información, más oportunidades habrá de que este innovador sistema informático llegue a hacerse realidad.

Animamos a los lectores de los BBSs a discutir las posibilidades de que este ordenador sea factible, pero al mismo tiempo nos gustaría estar al tanto de todo ello. Así que por favor enviad las ideas resultantes o las conclusiones por correo "normal" (no electrónico) a la dirección de arriba.

<*** Fin del mensaje ***>

Nederlando

Bonvolu pludoni cxi tiun mesagxon al lauxeble plej da uzantoj, grupoj revuoj kaj BBSoj (pri QL, ST, Amiga kaj Mac). Ju pli da homoj ni atingos per cxi tiu sciigo, des pli da ebloj realigi cxi tiun novigecan komputilan sistemon.

Ni kuragxigas la BBS-legantojn diskuti la realigeblecon de cxi tiu komputilo, sed samtempe ni volas resti informataj pri tio. Do bonvolu sendi la rezultajn ideojn aux konkludojn per normala posxto (neelektronika) al la supra adreso.

<*** Mesagxofino ***>

Notas sobre la traducción del inglés al castellano:

sófgüer = neologismo propuesto por mí para "software", logicial o lógical, en función del mismo proceso por el que ahora nos parece tan "natural" la palabra "fútbol", que hasta hace muy poco (décadas) aún no se escribía en los diarios, pues era muy chocante, sino el término del que procede, el británico "football", en cursiva. Otro proceso con resultado similar pero camino de adaptación distinto se produjo con "bricolage", "sidecar", "iceberg" y tantas otras. Es un proceso habitual de intercambio entre las lenguas, por una lado útil y por otro peligroso cuando la balanza pesa más de un lado que del otro o se produce sin necesidad real, pero en cualquier caso inevitable. El término deseable sería "logicial", pero me temo que su batalla está casi perdida, así que tarde o temprano pasará con "software" como con "football", como con "leader", como con "meeting"...

járgüer = más de lo mismo, con la diferencia de que en este caso no hay alternativa, a no ser que hagamos como ellos y usemos "ferretería", "hierrajos" o algo así...

Traducido del inglés al español y al esperanto por:
Elangligita hispanen kaj esperanten far: Marcos Cruz
Translated from English into Spanish and Esperanto by:

El rincón del pinchadiscos

Los últimos discos llegados a la colección son:

- C68 v3.01 runtime (3 discos).
- C68 librería QPTR (1 disco).

Como veis, es muy probable que el tesoro de QLíper pueda estar lleno de telarañas -sin un miserable ECU- , pero eso no significa que por mi parte deje de seguir aumentando la colección de programas de dominio público de nuestro querido club, especialmente si los programas son de mi interés personal.

Si alguno de vosotros desea consultar algo sobre la librería, o el mundo QL en general, podéis poneros en contacto telefónico conmigo llamando al (95) 2475043 en horas de trabajo y al (95) 2474887 cuando estoy en casa.

Salvador MERINO, 1992 07 04

Programas de dominio público

A continuación pasaré a hacer un breve comentario sobre algunos de los programas de dominio público que forman parte de la amplia biblioteca de Salvador Merino.

Antes que nada convendría señalar qué es lo que se entiende por programas de dominio público. Esta denominación se aplica a aquellos programas en los que sus autores renuncian a sus derechos de autor, por lo que pueden ser distribuidos y copiados libremente. Para ello, los canales usuales son las librerías de programas de dominio público, en las que el usuario paga una pequeña cantidad de dinero para cubrir los gastos del soporte físico, gastos generales de mantenimiento de la librería, gastos de envío, etc., y las bases de datos telefónicas (BBS). Por supuesto, otro medio eficaz de distribución es el de "mano en mano". Otros canales pueden ser revistas comerciales, clubes de usuarios (parecidos al nuestro) y alguna otra más que ahora no se me ocurre.

Además de poder ser copiados libremente, se pueden utilizar sin ninguna restricción. En algunos casos el autor facilita su dirección para que, aquellos que hayan sacado partido del programa en cuestión y lo deseen libremente, les envíen un pequeño donativo. Normalmente en estos casos el autor procuraría enviar la última versión actualizada del programa (si existe), aunque no está obligado a ello.

Otro modalidad similar es lo que se viene en llamar distribución pública ("shareware"). Aquí de lo que se trata es de, mediante unos canales de distribución iguales a los de los programas de dominio público, dar al usuario una oportunidad "probarlo antes de comprar". Si uno utiliza un programa de distribución pública habitualmente está obligado (moralmente al menos) a enviar una cantidad al autor en concepto de cuota de registro. De este modo, el autor se compromete a enviarnos una mayor documentación (normalmente un manual escrito) y a actualizarnos nuestro programa a medida que vayan apareciendo nuevas versiones.

Esta fórmula depende mucho de la honestidad de los usuarios, pero en algunos países os puedo asegurar que funciona. Como ejemplo, os diré que, aunque utilizo normalmente varios programas de este tipo para mi Atari o para PC, sólo he enviado esta cuota para dos programas. Se trata de dos juegos para Atari ST de un autor británico, que ha recibido más de 1000 peticiones de registro de todo el mundo, y que sigue recibiendo unas 20 diarias. Teniendo en cuenta que la cuota es de 5 libras, se puede decir que para él le ha sido más rentable poner su programa en distribución pública que haberlo puesto en el mercado como un programa comercial (y eso que él piensa que sólo un 1% de las personas que poseen estos programas se registran como usuarios).

Se puede considerar esta formula como un alternativa viable a las redes de comercialización convencionales, aunque el exito económico resulte difícil de obtener normalmente.

Como un consejo práctico, tened cuidado si os queréis registrar de un programa y éste es extranjero. Lo primero que hay que hacer es asegurarse de si el autor sigue estando en las señas que suelen venir en la documentación del programa y si sigue soportando el programa en cuestión. Para ello lo mejor es enviarle una carta con un sobre con nuestra dirección para la respuesta, junto con un vale de respuesta internacional. En caso de recibir respuesta, entonces podemos pasar a registrarnos.

Además de estas dos modalidades, existen otros tipos, como son las demos de programas comerciales (que siguen manteniendo el derecho de copia, pero que tienen deshabilitadas algunas funciones básicas, como el Salvar o Imprimir, o que sólo funcionan durante un periodo corto de tiempo), y otras que pueden ir surgiendo en el futuro.

En lo que se refiere al QL, creo que la inmensa mayoría de los programas son de dominio público. Sin embargo, los programas de la librería de Quanta se podían considerar como de distribución pública, ya que había que pagar a los autores (si bien en este caso antes de recibir los mismos, independientemente de si luego te parecían buenos o no), pero creo que esto ha cambiado en los últimos tiempos (debido a la dificultad que para mí entrañaba este sistema, nunca he utilizado la librería de Quanta, y este año no he renovado la subscripción).

Una de las ventajas de estas modalidades es que existe una libertad total de los autores a la hora de escribir sus programas, sin necesidad de consideraciones de orden comercial. De este modo, en estas modalidades se pueden encontrar desde pequeños programas que están pensados para resolver pequeños problemas muy especificos, por ejemplo un controlador ("driver") de impresoras poco usuales, hasta programas muy complejos (hojas de cálculo, bases de datos, etc), con unos niveles de calidad muy heterogéneos, pero que en algún caso pueden ser incluso superiores a muchos programas comerciales.

Os animo a que os introduzcáis en este mundo tan interesante. Seguro que encontraréis ese programa que tanto estabais buscando pero no encontrabias. Y si no, también habrá merecido la pena. Os lo aseguro.

Por lo que se refiere a la biblioteca de Salvador Merino os diré que es bastante amplia y que se incluyen cosas bastante interesantes. A continuación comentaré algunos de estos programas.

DISCO IMG GIF1.-0

En este disco se incluyen unas cuantas pantallas en formato GIF. Este es un formato muy popular en el mundo PC. Se trata de un formato comprimido, que guarda la imagenes independientemente de la resolución. Mediante el programa adecuado, estas imágenes se pueden visualizar en los diversos modos utilizados en los PC, desde CGA (320x200 pixeles con una paleta de 4 colores) hasta VGA (1024x768 pixeles con una paleta de 16 millones de colores).

Este formato fue creado por COMPUSERVE, una de las más importantes BBS en los Estados Unidos. Por cierto, se que en algunos países de Europa se puede acceder de una forma económica a este servicio, pero no se si en nuestro país es posible. Como no tengo módem no me preocupa mucho el saberlo por el momento, pero si a alguno de vosotros os interesa, su dirección es: COMPUSERVE Inc., P.O. Box 20212, Columbus, OH 43220-9988, EE.UU.

En cuanto a las pantallas en sí las hay de varios tipos más o menos vistosos. Lo más interesante es que se incluye un programa llamado SHOWGIF.EXE, que se encarga de transformar las pantallas en formato .GIF a pantallas QL. Para ello ejecutamos el programa tecleando "exec showgif_exe". El programa nos pedirá a continuación el nombre del fichero en formato GIF que queremos visualizar, teniendo que incluir el nombre del medio en que se encuentra.

Una vez leído el fichero, se nos presenta una información sobre el formato original de la pantalla (dimensiones en pixel, número de colores, paleta uitlizada, etc) y si queremos cambiar el modo sugerido para la conversión (modo 4 ó modo 8).

Una vez contestada a esta pregunta (mediante "Yes" o "No") el programa nos pregunta si queremos utilizar la paleta estándar del QL (8 ó 4 colores según el modo), o si queremos usar una paleta artificial, usando los patrones de puntos ("stipples"). Los resultados varían mucho según la imagen, por lo que lo mejor es probar hasta lograr el mejor resultado. Una vez hecha la conversión, el programa nos preguntará si queremos salvar la pantalla como una pantalla de QL. En caso afirmativo, le daremos el nombre del medio y del fichero.

No se trata de un gran programa, ya que no es muy rápido en su ejecución y además no es nada "amistoso". Cuando acaba de hacer la conversión de una pantalla el programa se termina, y, si queremos repetir el proceso, tendremos que volver a ejecutarlo. Si nos equivocamos en cualquier paso no es posible rectificar el error. Sin embargo, las conversiones son bastante buenas, más si tenemos en cuenta la limitada variedad de la paleta de colores de nuestro QL. No conozco otro programa de sus características para el QL. Es una lastima que en el disco no se encuentre el listado fuente, para que se pudieran hacer algunas modificaciones que le vendrían muy bien. En suma, un programa que podría tener bastantes mejoras, pero que permite a los usuarios del QL acceder a una vasta cantidad de pantallas del universo PC.

DISCO H007

Dentro de los programas incluidos podríamos destacar los siguientes:

UNZIP96_EXE: Programa holandés, ejecutable en multitarea. Este programa sirve para extraer ficheros comprimidos en formato ZIP, formato originalmente desarrollado para el sistema operativo MS-DOS. Mediante el comando exec_unzip96_exe o mediante EX/EW (si se dispone del TKII) con una cadena para seleccionar las distintas opciones el programa comienza su ejecución. Para comprender el significado de las distintas opciones se incluye documentación en inglés, disponiendo también de información en línea.

Como hemos dicho, este formato ZIP fue desarrollado para el MS-DOS para la compresión de ficheros, de modo que se pudiera ahorrar espacio (y tiempo) en las transmisiones de estos ficheros via modem. Este formato se fue extendiendo a otros sistemas, y los programas de comunicaciones de otros sistemas (Mac, Atari, Amiga) suelen soportar este formato. También es usual en los discos de dominio público utilizar este formato para poder "meter" más información en disco.

En el mismo disco se incluyen otros programas dedicados al mismo tema, como el QZ_EXE y UNARC_EXE, que abarcan otros formatos, también muy extendidos, siendo muy sencillo el utilizarlos. En la misma línea y tratando específicamente al QL se incluyen los programas COMPACT_EXE y UNCOMPACT_EXE. En suma, una serie interesante de programas, dedicados especialmente a aquellos interesados en el tema de las transmisiones via módem.

HARDBACK_EXE: Programa de copia de respaldo o de seguridad ("back-up"). Este programa está pensado para hacer copias de seguridad de disco duro a diskettes, pero se puede utilizar con otro tipo de dispositivos, como discos RAM, o hacer copias de diskette a diskette. Se trata de un programa ejecutable en multitarea, con información en inglés. Una vez comenzada la ejecución del programa (mediante el comando EX HARDBACK_EXE), éste nos pregunta acerca del directorio fuente y el destino. A continuación nos pregunta si queremos hacer las copias de todos los ficheros o sólo de aquellos que estén actualizados desde la fecha de la última copia de seguridad. Esta opción es de particular interés para aquellos que posean un disco duro, ya que así se facilita extraordinariamente la pesada tarea (pero por otra parte tan importante si queremos tener a buen recaudo nuestros valiosos programas) de hacer copias de seguridad. Es un programa sencillo de usar y de gran utilidad para aquellos que dispongan de disco duro.

En suma, un disco con una serie de programas de calidad, sencillos de usar y utiles. En próximas ocasiones volveré a comentar alguno de los programas que están al alcance de todos nosotros. Animaos a usarlos, ¡merece la pena!

Javier ZUBIETA, 1992 06 07

¡Quiero escribir en QLíper!

Pues sí, tengo muchas cosas que contaros y poco tiempo para hacerlo. Se me ocurren tantas cosas que no las puedo sacar a la luz. Por eso estoy escribiendo a toda prisa esta pequeña reseña indicando en qué áreas de interés me estoy moviendo y según vaya terminando en alguna os iré pasando lo que haga. No sé cuándo, pero irá saliendo. Os lo presento ahora porque me muero de ganas de enseñaros lo que estoy haciendo y no puedo esperar a tenerlo terminado. ¿Me comprendéis, verdad, compañeros?

DeskJet 500:

Una magnífica impresora, cada vez más barata. Llevo casi dos años usándola y va perfectamente. Ya hay muchos programas para QL que están adaptados para ella. Una buena elección si tenéis que cambiar de impresora. Hay una opción en color y HP sacará otro modelo en color, más rápido, dentro de poco.

Coherent:

Por 14000 ptas he comprado este sistema operativo que es un clónico de Unix para PC. Lo he instalado en mi PC portátil y estoy encantado. ¡Es Unix de verdad! Requiere sólo 10 M de disco duro y 1 M de RAM, y viene con un montón de herramientas típicas del mundo Unix. El manual es de 1200 páginas. Es una de las mejores maneras de usar un PC, y tiene un precio tirado. En el artículo

"Coherent" podéis ver por encima las posibilidades del producto. El texto lo he sacado directamente del sistema en formato MS-DOS y lo he importado al QDOS con Xover; un poco de edición con The Editor y ¡ya está!, limpito. Si deseáis más información (antes de que escriba un artículo), podéis pedirla a: Mark Williams Company, 60 Revere Drive, Northbrook, IL 60062, U.S.A.

Informática portátil:

Los ordenadores portátiles cada vez son más baratos, potentes y pequeños. Empiezan a aparecer aparatos muy pequeños (A5) y ligeros (1Kg) con suficiente potencia como para poder llevar mucha información en el bolsillo (40 M de disco). Un buen ejemplo es el HANDBOOK, de Gateway 2000, pero hay muchos más. Cuando los chips de 3V estén disponibles a buen precio veremos equipos aun más potentes y pequeños que estarán alimentados por 2 pilas AA y darán 8 horas de autonomía real, al menos. Apple presentará su nueva gama de DPA (Digital Personal Assistant) con la intención de revolucionar el marcado. ¿Para cuándo el QL portátil?

Programación orientada al objeto en SuperBASIC:

Estoy montando un conjunto de rutinas reutilizables, orientadas al objeto, que permitan reducir mucho el tiempo de creación de un programa. La estructura general ya la tengo, muchas rutinas también, pero falta probarlas y hacer un primer programa que las utilice para poder ponerlo como ejemplo. El programa se llamará Mondrian y servirá para definir las ventanas que usará un programa; es decir, el primer programa que voy a hacer servirá también para hacer programas. La pescadilla viciosa que se muerde circularmente la cola... En el fichero POOSB_txt [Red.: incluido como artículo en este número] podéis echar un vistazo a las líneas maestras de lo que estoy haciendo.

Programación en C:

Ya os presenté un ejemplo de programa compilado en C68 usando unos ficheros de cabeceras que nos facilitan enormemente el acceso a las funciones de C. El programa se llamaba DemoGraf. Las cabeceras que usaba están pensadas para C68 v2.00 y no funcionan correctamente con C68 v3.01. Mi idea es reescribir los ficheros de cabecera para poder escribir programas en C que se puedan compilar (cambiando el código lo menos posible) en tres sistemas operativos distintos: QDOS, COHERENT y MS-DOS.

Después llevaré las ideas de programación orientada al objeto que estoy desarrollando en SuperBASIC a C y, teóricamente, podremos escribir programas de cierta entidad con rapidez y podremos "exportarlos" a otros sistemas. ¿Lo conseguiremos?

QSUP:

Un magnífico conjunto de herramientas y extensiones ("toolkit") vendido por Jochen Mertz que me permite olvidarme casi por completo de los problemas de traducción de caracteres al imprimir. Ya sabéis: cómo conseguir que los caracteres del QL se trasvasen correctamente a la impresora.

Cifras:

¿Habéis visto el programa "Cifras y letras" de TVE?. Pues he hecho un programa en C que resuelve la parte de las cifras. El programa ya está funcionando (y bien) bajo MS-DOS, pero no lo podéis ver en el QL porque el montador de C68 no es capaz de montar el ejecutable. Cuando llegue la solución del equipo que lleva C68, os lo pasaré.

WSET:

Después de muchos años intentándolo he conseguido escribir una "cosita" en ensamblador. Es una extensión del QDOS que se llama WSET y permite redefinir las ventanas 0, 1 y 2 del SB a nuestro gusto. Cuando le prepare un programa en SB para personalizar la extensión, lo veréis.

Fuente del PC:

El QL no tiene símbolos semigráficos (ni falta que le hace), pero son una buena

manera de hacer cuadros en una impresora. Para poder manejar los cuadros con comodidad es conveniente usar una fuente de letras que tenga semigráficos. Ya puestos, como casi todas las impresoras tienen el conjunto de caracteres del PC (se llama PC-8 en mi DeskJet 500), es buena idea tener una fuente de letras en formato QL con los caracteres PC. Ya la tengo, pero la publicaré cuando os prepare un programa para crear cuadros fácilmente.

KandiPlus

Ya tenéis la versión actual del programa Kandi, que me permite generar gráficos mediante fórmulas matemáticas. He preparado una versión que produce los gráficos a toda pantalla, para poder apreciar mejor las posibilidades. Seguramente en el próximo número de QLíper aparecerá.

Pedro REINA

Tarjeta "QL EMULATOR" para Atari ST: primeras impresiones

Cuando adquirí mi entrañable, alargado y negro QL en Noviembre de 1985 no tenía una idea muy clara de qué me depararía el futuro.

En el año 1985 el QL se presentaba como una alternativa muy interesante. Un procesador de 32 bits (aunque tuviera un bus de datos de 8 bits, debemos tener en cuenta que todavía hoy la mayoría de PCs tienen unos procesadores de 16 bits e incluso las versiones llamadas SX de los procesadores Intel, que tienen un bus limitado a 16 bits, aunque su estructura interna sea de 32 bits en los modelos 80386 y 80486, son muy populares en el mundo de los clónicos), un sistema operativo multitarea y una memoria de 128 Ko, junto con unos "microdrives" mucho menos engorrosos que las unidades de caset, eran características muy avanzadas e interesantes. Además, la bajada de precios de 125.000 a 79.500 pts lo hacía más interesante todavía.

Con el correr del tiempo y cuando mi situación económica me lo permitió, al empezar a trabajar, empecé a adquirir diversos complementos para potenciar mi ordenador. Primero fue una ampliación de memoria Sandy SuperQboard. Luego fue una unidad de disco de 3 1/2" también de Sandy y luego una impresora Epson LX86. Dada la nula posibilidad de adquirir estos elementos en suelo patrio tuve que acudir a la compra de los elementos específicamente para QL vía venta por correo a Inglaterra, afortunadamente sin muchos problemas. La compra de estos elementos transformaron verdaderamente mi QL, haciéndolo una máquina mucho más capaz y potente. Con 640 Ko de memoria y el TK2 de Tony Tebby las estupendas características multitarea del QL podían ser explotadas a plena potencia.

Pero el tiempo pasa inexorablemente y nuestro QL iba languideciendo poco a poco, sin ningún apoyo oficial en nuestro país. Después de dos averías, a las cuales creo que ya me he referido en estas mismas páginas, una terrible angustia se apoderó de mí. ¿Qué había hecho yo para merecer esto? cada vez que a mi QL se le ocurriera estropearse un sudor frío recorrería mi espalda, me pondría a soltar algunos improperios y encendería alguna vela a la Virgen de Lourdes para que esta avería pudiera ser facilmente reparada por mí o por algun taller que, por pura casualidad, tuviera a bien realizar la reparación. Además tenía la sensación de estar completamente aislado, al no conocer aún a nadie que tuviera un QL. Menos mal que en esto descubrí CUQ y desde entonces vi las cosas con un poco más de optimismo.

Con este panorama, comencé a ver la posibilidad de pasar a un sistema más potente. En principio descarté los PCs, al no gustarme mucho el MS-DOS y su precio, prohibitivo para mí en aquel entonces (aunque luego volveré a referirme a este punto). Para poder aprovechar en algo mis escasos conocimientos de ensamblador, mi idea era la de pasarme a un sistema construido alrededor de la familia 68000. Si además este nuevo ordenador podía emular al QL, entonces mejor que mejor.

Así las cosas, mi primera idea fue adquirir un Commodore Amiga. Sin embargo el modelo básico, el Amiga 500, es un tanto limitado y el superior, el A2000, excesivamente caro. Su ventaja mayor es que existía un emulador del QL en el dominio público. Posteriormente pasé a considerar el Atari ST. Los modelos 520ST y 1040ST me parecían también limitados y los modelos "profesionales" Mega

eran excesivamente caros. Para realizar la emulación del QL en estos modelos existía una tarjeta que costaba unas 200 (unas 38.000 pts).

Finalmente me decidí por el modelo Atari Mega ST2 en abril del año 90, al coincidir una oferta (150.000 pts+IVA). Mi idea en un principio fue la de adquirir lo antes posible la tarjeta emuladora de Jochen Merz Software, pero fui retrasando la decisión, al aparecer en esos momentos una nueva versión de la tarjeta, la llamada "extended mode 4", que no incorporaba la ULA ZX8301 (lo cual es una ventaja añadida, al ir siendo cada vez más difícil encontrar repuestos originales para el QL, al haberse dejado de fabricar a comienzos de 1986) y que presenta la posibilidad de tener un modo 4 con una resolución de 768X280 pixeles (más de un 50% de información adicional en pantalla), aunque tiene el inconveniente de que no dispone del modo 8.

Tras consultar en varias ocasiones a Jochen Merz y enviarle una imagen de la ROM MGE de mi QL, además de la disposición del teclado de mi Atari, el pasado diciembre me decidí por pedir la versión QL-Emulator Extended 8, ya que prefería mantener la máxima compatibilidad posible. Mi primera sorpresa fue ver que el precio de esta tarjeta había descendido considerablemente, pasando de las 38.000 pts a 21.760 pts (327.45 DM + 12DM de gastos de envío). La razón es que este modelo va a desaparecer en beneficio del QL-Emulator Extended 4.

Junto con la tarjeta, que tiene una dimensiones muy reducidas, se incluye un manual que explica cómo usar las extensiones del Hotkey System II, además de las instrucciones de montaje de la tarjeta. Para el mismo se debe tener en cuenta el modelo concreto de ordenador de que disponemos, ya que hay bastantes diferencias entre las placas bases de las distintas versiones de los STs.

Básicamente se trata de insertar la tarjeta en el espacio ocupado por uno de los chips del ST, concretamente del chip cambiador ("shifter") de vídeo. Según el tipo de placa base, habrá que desoldar este chip o simplemente sacarlo de su zócalo. En mi caso el chip estaba soldado. Como no me atreví a desoldarlo lo tuve que llevar a un servicio técnico, donde por cierto, que cobraron a precio de oro el desoldar este circuito y el uP 68000 (12.000 pts + IVA). Una vez puesta la placa en el lugar del chip cambiador ("shifter") se tiene que soldar 7 cables a distintos lugares de la placa.

La parte mas complicada es sacar el chip llamado "glue" de su zócalo de 64 patillas de forma cuadrada y soldar directamente a dos de sus patas dos cables, doblando los contactos del zócalo para que no haya conexión entre el zócalo y

esas dos patas. Precisamente el no entender bien esto fue lo que motivó que no pudiera ser capaz de instalar correctamente la placa, lo que me obligó a tener que mandarlo a Alemania para que me lo instalaran.

Lo más curioso (¿será mi sino?) fue que cuando lo envié por EMS (un servicio rápido de Correos) fue un viernes y al lunes siguiente comenzó la primera huelga general en los servicios públicos en Alemania en 18 años. La broma me costó unas 20.000 pts (9.000 pts del porte España-Alemania, 48 DM de instalación y 130 DM del porte Alemania-España). Y todo por una mala interpretación mía de la instalación, que una vez vista (vaya mérito el mío) es muy sencilla. Total, que la broma me salió por unas 50.000 pts. La verdad es que con un poco más de habilidad por mi parte me podía haber salido por la mitad, pero qué se le va a hacer.

Una vez instalada la tarjeta, el Atari se comporta como tal, hasta que se le hace arrancar con un programa incluido con el emulador, en formato ST. Una vez que se ejecuta este programa de arranque, en la pantalla aparece la habitual pantalla que nos indica que pulsemos las teclas F1 o F2, según utilicemos un monitor o una TV, respectivamente.

Si lo deseamos podemos hacer que el Atari arranque como QL sin introducir el disco de arranque que viene incluido con la placa. Si lo hacemos así, tendremos un QL con practicamente toda la memoria de que disponga el ST, pero "básico", es decir, sin TK2, con el teclado configurado como un ST alemán, y sin la posibilidad de cambiar del modo 4 al modo 8. Esto puede ser conveniente para utilizar algunos programas que de otra manera "se cuelgan", sin que sepa, de momento, la causa.

Lo normal es arrancar con un disco de arranque que cargue las extensiones

incluidas con el disco suministrado por Jochen Merz, que proporciona las extensiones del Hotkey System II. Aunque no he tenido tiempo de experimentar con estas extensiones, puedo deciros que son un desarrollo del sistema QRAM, pero con más opciones.

El poder utilizar este entorno, con una mayor velocidad , con mayor memoria (1.6 Mo en mi caso) y con un ratón, es algo estupendo, que no cambio por el Windows para PC (al menos que no sea en un PC con un 486 a 33Mhz y disco duro de 100 Mo).

En cuanto al funcionamiento general, aún no he tenido mucho tiempo para probarlo, pero parece que el sistema es bastante compatible. De cualquier manera, hay que recordar que no existe ninguna emulación que sea 100% compatible. Dentro del mismo mundo QL, donde no ha habido una gran evolución del mismo debido a los problemas que todos sabemos, tambien hay (ligeros) problemas de compatibilidad entra las distintas versiones de ROM. Tambien en el caso del Atari hay distintas versiones del sistema operativo, con sus ligeras incompatibilidades (y gazapos).

Por lo que se refiere a los programas de Psion todos parecen funcionar correctamente, si bien el doble de rápido, lo cual los hace un poco más atractivos, ya que, para los estándares de hoy, ¡hay que reconocer que son lentísimoooooossss!

El teclado responde bien, pero de momento no he conseguido que manipule correctamente los acentos. Para sacarlos tengo que utilizar la tecla CTRL combinada con otra tecla, lo cual resulta algo confuso. Cuando tenga algo de tiempo le consultaré al Sr. Merz cuál es el problema (me imagino que debo de ser el único usuario español que ha adquirido esta tarjeta). Otra particularidad es que el emulador está basado en la ROM JS (inglesa), aunque se pueden obtener los mensajes de error en castellano con una extensión que viene en el disco del

emulador. Una curiosa consecuencia de esta versión de la ROM es que no permite que las variables de SuperBasic incorporen la letra "ñ", si bien este no es un problema excesivamente grave.

Una de las ventajas más apreciables que veo en mi sistema es que el teclado de mi Mega ST2 es mucho mejor que el de mi QL, y que la fiabilidad de la fuente de alimentación del Atari es infinitamente superior a la del QL (es rarísimo que el Atari se quede colgado por alguna interferencia en la línea eléctrica, mientras que el QL se queda colgado en cuanto alguien tose cerca del enchufe en el que está enchufado).

Mi idea es adquirir un disco duro en breve, para poder compartirlo entre el Atari y el QL, si bien tengo el eterno problema: el Atari esta languideciendo en España a gran velocidad y es casi imposible encontrar un disco duro a un precio razonable. Bueno, ya os hablaré en ocasiones venideras de mis visicitudes con mi nuevo sistema. Mi intencion es, si todo va correctamente, desprenderme de mi viejo QL para poder hacer un poco de sitio en mi mesa, ya que ahora la tengo llena de cachivaches que me impiden cualquier movimiento cómodo. Como reflexión final os diré que hasta ahora estoy satisfecho de mi adqusición (a pesar de los problemas que he tenido) y os animo a que sigáis mi camino, si os interesa, más ahora que acabo de recibir el ultimo folleto de Jochen Merz, donde se señala que se puede utilizar esta tarjeta con placas aceleradoras basadas el los 68020 y 68030, con frecuencias de hasta 50 Mhz y coprocesadores matemáticos. Yo tenía entendido que el QDOS no podía funcionar con estos procesadores (ya que éstos asumen que el código debe ser "puro", no admitiendo el código automodificable y algunas pequeñas chapuzillas que sí admite el 68000).

También quiero comentar algo a lo que me había referido al comienzo. Hace 7 años, cuando adquirí mi QL, el precio de un PC decente, para usarlo como ordenador "casero", era prohibitivo. Pero hoy en día, hay incluso casas que venden PCs en plan "bricolage". Te venden por separado placas madre (por ejemplo, por 20.000 pts puedes comprar una placa madre con un 80386SX a 16Mhz), fuentes de alimentación, tarjetas de vídeo, cajas para el montaje, discos duros, etc., a precios increíblemente bajos, pudiendo pedir que te monten el ordenador a tu medida, o hacerlo uno mismo con estos elementos (se puede tener un ordenador bastante decente por menos de 90.000 pts). Con este panorama, no creo que los otros sistemas puedan resistir mucho tiempo. Me temo que en un plazo de 3 ó 4 años no me quedará más remedio que pasarme a un PC. Bueno, tal vez esto no

sea tan terrible. A veces me pregunto qué tendrá esto de la informática, que hace que uno le tenga tanto amor a su ordenador como a su equipo de fútbol, pasando a ser un furibundo hincha, cuando se supone que los "entendidos" en informática deberíamos ser racionales y cerebrales.

De cualquier modo, por mi parte os brindo mi pequeña experiencia a quien pudiera estar interesado en este emulador.

Javier ZUBIETA, 1992 06 07

Oferta

Tras casi siete años de relación amistosa he decidido poner a la venta mi querido y viejo, aunque todavia en plena forma QL. ¿Signifa esto que abandono el barco, cual mísera rata? ¡De eso nada! Lo que sucede es que, como tal vez sepáis por artículos míos, tengo desde hace unos meses una tarjeta emuladora QL-EMULATOR instalada en mi Atari Mega ST2. Dado que estoy muy satisfecho con esta combinación y ultimamente tengo arrinconado mi QL, he decidido ponerlo a la venta, para sacar así unas pesetillas y librar un poco de espacio en casa. Así que, si alguno quiere tener un segundo QL como repuesto, o conocéis a alguien

interesado en introducirse en el maravilloso mundo del QL, esta es vuestra oportunidad. La oferta que pongo a vuestro alcance es la siguiente:

a) QL, MGE, comprado en noviembre de 1985, con ULA ZX8301 cambiada hace poco, en buen uso, incluyendo uP 68008 y 8049 de repuesto. Incluyendo fuente de alimentación, cable para conexión para TV y cable para Euroconector. Se incluyen también los programas de Psion. No se incluye el manual (¡ojo!). Embalado en su caja original. Precio: 15.000 pts.

b) Placa SANDY SUPERQBOARD, con 512 Ko de RAM, TKII, puerto paralelo para impresora y puerto para conexión unidad de disco (5 1/4 ó 3 1/2"). Se incluye cable centronics para impresora. Precio: 10.000 pts.

c) Conjunto a) + b): Precio: 20.000 pts.

d) Conjunto de libros: SINCLAIR QL ADVENTURES, Ed. SUNSHINE (en inglés); ARTIFICIAL INTELLIGENCE ON THE SINCLAIR QL, Ed. SUNSHINE (en inglés); MATHEMATICS ON THE SINCLAIR QL, Ed. SUNSHINE, (en inglés); DEVELOPING APPLICATIONS ON THE SINCLAIR QL, Ed. SUNSHINE (en inglés); QL. LIBRO DE JUEGOS, Ed. RAMA, (en castellano); MANUAL DE REFERENCIA PARA EL SINCLAIR QL, Ed. RAMA, (en castellano). Precio de todo el lote (no se vende por separado): 2.000 pts.

e) Programas en "microdrive": TANKBUSTERS, WANDERER, QL FLIGHT, KARATE, DEATHSTRIKE, THE LOST PHARAOH, ZAPPER & EAGLE, 3D SLIME, CARTRIDGE DOCTOR, MATCH POINT, PSION CHESS, DROIDZONE, JUNGLE EDDI, 4MATTER, GENERADOR DE SPRITES DE DIGITAL, LOCKSMITHE, COPYCAT, COPIER, ALIEN HIJACK. Todos los programas son originales (por cierto, agradecería que alguien me dijera cómo desproteger el CHESS y el MATCHPOINT, para poderlos utilizar en disco con mi Atari+QL-EMULATOR). Precio: 500 pts por unidad. Todo el conjunto por 7.500 pts. (incluyendo dos cables adaptadores joystick).

f) Ratón Starmouse + programa. Precio: 1.500 pts.

g) Cartuchos de "microdrive": 45 ud. Precio : 200 pts unidad. Oferta de todo el conjunto: 35.000 pts.

Forma de envío: a convenir. Gastos de envío: aparte. Espero vuestras ofertas. Por favor, hacedlo por correo. Mi dirección es: Javier Zubieta Aguirre, Novia Salcedo 26-4º-izda., 48012 BILBAO

Repaso al mundo PC o "no es oro todo lo que reluce"

¡Que no cunda el pánico! Ya sé que este no es lugar para hablar y alabar a esas máquinas no muy queridas por nosotros, los QL maniáticos. Pero creo que

convendría hacer algunos comentarios acerca de estos ordenadores, ya que los tenemos hasta en la sopa y todos, quien más y quien menos, se ha tenido que enfrentar con ellos, ya sea en el trabajo, en la universidad, etc.

Sin querer ser muy rigurosos, podemos hacer una breve historia acerca de estos ordenadores. Creo que aparecieron en el mercado en 1981, tras serle levantada a IBM una especie de imposición o compromiso por el cual no podía fabricar mini-ordenadores (entrando en esta categoría todo lo que no fueran "main-frames", o sea, ordenadores grandes, según el criterio de la epoca. Hoy hay ordenadores "personales" e incluso "domésticos" mucho más potentes que la mayoría de los grandes ordenadores de aquella época).

En su primera aparición, los PCs venían con 64 Ko de memoria RAM, creo que 48 Ko de memoria ROM y tenían incluso un conector DIN para utilizar una unidad de caset (sí, como si de un Spectrum se tratara). No contaban con capacidad gráfica. Su UCP era un microprocesador 8088, que corría a 4.77 Mhz. Este microprocesador es de 16 bits, aunque con un bus de 8 bits. Era el sucesor del uP de 8 bits 8080, del cual se derivó el Z80. La verdad que estos PCs originales no tenían una gran ventaja sobre los otros ordenadores vigentes más populares, en aquella época, que solían incorporar un uP Z80 (como el Spectrum) y utilizaban el sistema operativo CP/M. Sin embargo contaban con una serie de ventajas innegables. En primer lugar, estaban avalados por el fabricante más importante y con más prestigio del mundo. Además, los diseñadores tuvieron la brillante idea de diseñar un sistema abierto, al incorporar en la placa madre una serie de ranuras ("slots") libres, que irían permitiendo incorporar tarjetas que, desarrolladas por otras compañías o por la propia IBM, permitirían el ir incorporando nuevos desarrollos en "hardware". Además de concebir el sistema como abierto, IBM tuvo desde el principio la idea de permitir que otras compañías desarrollaran ordenadores basados en la misma arquitectura, los famosos "compatibles". El éxito de esta medida lo conocemos todos. Incluso para aquellos a quienes no nos gustan los PCs, debemos admitir que es el estándar "de facto" en el mundo de los ordenadores personales. Si todos los fabricantes de ordenadores hubieran tratado de ponerse de acuerdo en imponer un estándar no hubieran sido capaces (recordad el sistema MSX, que, por cierto, se sigue vendiendo en Japón). La realidad nos enseña que para que una norma se acepte a nivel internacional lo que hace falta es que alguien la imponga y lo demás son tonterías (las normas alemanas DIN, que tratan de casi todo lo imaginable, son las normas que, en la práctica, se imponen en Europa, dándose la circunstancia de que algunos países que tienen su propio instituto de normas, se limitan a copiar prácticamente las normas DIN, para incorporarlas a sus propias normas).

Al lanzar este ordenador, IBM apostó por un nuevo sistema operativo, rompiendo con el CP/M. Para ello contrató los servicios de una pequeña compañía fundada por un tal Bill Gates: Microsoft. La verdad que a la hora de diseñar este nuevo operativo, los chicos de Microsoft no tuvieron unas miras muy altas. Dado que el PC estaba desarrollado a partir del uP 8088, las limitaciones de éste limitaron mucho el potencial del nuevo sistema operativo, el MS-DOS.

El 8086 es un procesador de 16 bits, como ya habíamos dicho antes. Fue lanzado al mercado en 1979 por Intel. Tenía unos 27000 transistores y venía en un chip de 40 pines (los mismos que, por ejemplo, los uP de 8 bits 6502 de Rockwell del Commodore 64, y el Z80 del Spectrum). Por ello, tenía que multiplexar el bus de datos. Para no ponerse muy técnicos, se puede decir que esto significa que este procesador no accede a la memoria por palabras de 16 bits, sino por octetos, lo cual ralentizaba su velocidad de ejecución si lo comparamos con un uP de 16 bits reales, como, por ejemplo el 68000 de Motorola.

Este procesador precisaba de unos circuitos especiales para decodificar sus señales, lo cual lo hacía caro para su época. Por ello, IBM optó por su hermano menor, el 8088, de ocho bits. Veremos luego que esta es una pequeña "chapucilla" que Intel ha practicado con asiduidad.

El 8088 o su hermano mayor, el 8086, que era prácticamente igual, sólo podían direccionar 1 Mo, pero lo hacían de manera segmentada. Esto significa que dividen la memoria en páginas de 64 Ko, lo cual los hacen bastante difíciles de programar, al menos desde el punto de vista de quien está acostumbrado a el direccionamiento lineal de la familia 68000. Sobre esto admito toda serie de discrepancias, ya que sólo estoy expresando mi opinión personal.

Con estas premisas, los programadores de Microsoft impusieron una serie de

limitaciones al MS-DOS, que prácticamente han permanecido inmoviles hasta hoy mismo.

El primer lugar, pensaron que un Mo era una cantidad de memoria inmensa, luego hicieron el mapa de memoria de su S.O. con este límite, asignando una memoria RAM máxima de 640 Ko. Esta limitación permanece hoy, si bien hay métodos para resolver esto, mediante la llamada memoria extendida y expandida, pero el MS-DOS sigue sin poder manipular convenientemente esta memoria extra, teniendo los programadores que "engañar" al MS-DOS para poder hacer uso de la memoria extra.

También pensaron que 32 Mo para un disco duro eran más que suficientes, por lo que limitaron a esta cantidad de memoria la capacidad de un disco duro formateado bajo el MS-DOS. Con la aparición de discos duros de grandes capacidades a bajo precio, se tuvo que parchear esta limitación dividiendo el disco duro en partes, llamadas "particiones" con un máximo de 32 Mo. Esta limitación ha permanecido hasta la reciente aparición de la versión 5 del MS-DOS.

Debido a la forma en que se trataba la memoria, la memoria de pantalla se trató en forma de caracteres, en lugar de linealmente (orientado por bits o pixeles), como en el caso de los ordenadores de la familia 68000 (QL, Macintosh, Atari, Amiga).

Esto tenía la ventaja de que permitía almacenar la información de la pantalla (recordemos, sólo de texto) fácilmente y ocupando poco espacio (80 caracteres x 25 filas = 2000 octetos). Esta ventaja se vuelve desventaja a la hora de las aplicaciones gráficas, aunque eso, en un primer momento, importaba poco.

En cuanto a la velocida de reloj, sólo podía trabajar hasta un máximo de 8 Mhz, aunque hay algunos uP compatibles, como la serie V20 de NEC que son un híbrido entre este uP y el 80186 (un uP apenas utilizado) que tal vez puedan trabajar a frecuencias mayores, pero que no creo que puedan pasar de los 16 Mhz.

Durante unos años, el mundo del PC no sufrió grandes cambios. El PC se fue introduciendo poco a poco, pero sin grandes alardes. Desde el punto de vista técnico, las novedades fueron apareciendo lentamente en un principio.

En 1982 Intel sacó al mercado su uP 80286, verdadero uP de 16 bits. Este microprocesador tenía un bus de datos de 16 bits y un bus de direcciones de 24 líneas, lo que le permitía acceder a una memoria de 16 Mo (la misma que la del 68000). Contaba además con la capacidad de gestionar memoria virtual. De este modo, el uP podía extender su memoria de trabajo a 16 Go, tomando memoria de dispositivos de almacenamiento tales como discos duros, como si de RAM se tratara, lo cual es una gran ventaja, de la cual han tenido que sacar partido los PCs para soslayar las limitaciones del MS-DOS. Integra 120.000 transistores.

Los diseñadores de Intel tuvieron cuidado extremo de mantener la compatibilidad ascendente con sus predecesores. Para poder hacer esto y, a la vez exprimir al máximo las nuevas características de este procesador, incorporaron dos "modos" de trabajo. En primer lugar estaba el modo "real" o modo 8086, y el modo protegido. En este modo se permitía el direccionamiento lineal de toda la memoria, al modo que estamos acostumbrados en la familia 68000. La velocidad de reloj de estos uP debe estar limitada a un maximo de 25 Mhz.

Con la aparición de este nuevo uP, en 1984 (aproximadamente), se introdujó el PC llamado AT. Las primeras versiones llevaban un 80286 trabajando a 6 Mhz, lo que hacía que fueran entre dos y cuatro veces más rápidos que un PC estándar. Sin embargo, para mantener la compatibilidad, las casas de programación hicieron que este uP trabajara práctimanente en modo real, es decir, como si se tratara de un 8086 "rápido", sin apenas sacar otra ventaja de sus características superiores, si exceptuamos la memoria virtual (si os fijáis en el funcionamiento de un PC observaréis que estan continuamente accediendo al disco duro, sobre todo si se está trabajando con aplicaciones que precisan de mucha memoria, ya que están supliendo la falta de memoria RAM por la memoria virtual del disco duro. Tal vez algunos hayáis pasado por la experiencia de que el PC os haya dado un "error fatal" al trabajar en un programa "devorador de memoria" cuando el disco duro estaba prácticamente lleno).

La mayor potencia de este uP permite la multitarea, con lo que se empezaron a implementar versiones del Unix para los AT (para los PC también se implementaron

versiones Unix, pero eran tremendamente lentas e inadecuadas para el trabajo encomendado). Sin embargo, el MS-DOS siguió siendo monotarea.

Aunque en principio los AT resultaban bastante caros, la aparición de este modelo supuso el despegue de los PCs. Hoy por hoy, es probable que el mayor número de ordenadores compatibles lleven un uP 80286, aunque hoy en día estan quedando obsoletos, habiéndose eliminado este modelo del catálogo de fabricación de Intel, y yo os recomendaría que si estáis pensando en comprar un PC o tenéis que asesorar a alguien, DECIDLES QUE NO COMPREN UN PC CON UN 80286 BAJO NINGUN CONCEPTO. Ya explicaré los motivos más adelante.

Posteriormente, en 1985, Intel lanzó su uP de 32 bits, el 80386. Manteniendo la compatibilidad ascendente con la familia 8088 se carateriza por tener una arquitectura de 32 bits reales. Es un dispositivo de alta escala de integracion, con unos 275.000 transistores. Tiene una capacidad de direccionar 4 Go de memoria principal y 64 To de memoria virtual, disponiendo de direccionamiento por segmentación y lineal.

Además del modo real y protegido del 80286, dispone del modo virtual o modo "8086", en el cual el 80386 puede trabajar como varios 8086 a la vez, en multitarea. Dispone de una pequeña cantidad de memoria caché (64 octetos). Su velocidad de trabajo puede llegar hasta 40 Mhz.

La aparición de este procesador en los PCs no se produjo hasta 1987-88. A pesar de su potencia, los fabricantes pronto se dieron cuenta de que fabricar PCs basados en una arquitectura de 32 bits no resultaba barato. Así que, para facilitar las cosas, Intel lanzó al mercado una versión "reducida", con un bus de datos de 16 bits, pero con la misma estructura interna. Así nació la version 80386SX, pasando a llamarse la versión de 32 bits 80386DX. La velocidad máxima de reloj de la versión SX es de 25 Mhz.

Dado que la tendencia actual de los PCs es trabajar en el entorno gráfico Windows (similar al entorno gráfico de ventanas e iconos que Apple introdujo en 1983 con su Lisa y posteriormente con el Mac) y este entorno precisa, para su mayor aprovechamiento (multitarea por ejemplo) el modo protegido, y que los precios de los PCs con procesador SX están por los suelos, la elección mínima de partida de alguien que quiere hacer algo serio con un PC es uno equipado con un 80386SX a 20 Mhz mínimo.

La continua aparición de nuevos procesadores Intel ha tenido como consecuencia la bajada inmediata de precios de la version anterior. Esto es particularmente cierto desde la aparición de los modelos 386SX.

Pero como Intel no se duerme el los laureles y sabe que, aunque la aparición de un nuevo modelo de uP no se asimila rapidamente, pero tiene un efecto inmediato en el incremento en la venta y abaratamiento de la versión anterior, vuelve a lanzar en 1988 un nuevo modelo, el 80486.

Este modelo, el último hasta ahora, aunque está anunciado para el año que viene el 80586, incorpora una UCP 80386 con un co-procesador matemático 80387 y un controlador de memoria caché en el mismo chip. Su velocidad de trabajo máxima es de 50 Mhz, aunque recientemente se ha lanzado una version que puede trabajar internamente a 66 Mhz, si bien el incremento de velocidad respecto a uno de 33Mhz es del 50% como máximo. El 80486 incorpora 1.200.000 transistores y tiene una memoria caché interna de 8 Ko. Se trata de un uP realmente potente, compatible en sentido ascendente con los modelos precedentes de su misma familia. Aquí también existe la version SX, que no contiene el co-procesador matemático, aunque el bus es de 32 bits y no de 16, como en el caso del 80386SX.

Aunque en principio, como suele ser habitual, los precios de los PCs basados en el 486 eran muy altos, ahora mismo se han abaratado mucho, no mereciendo la pena comprar un PC basado en el 80386 a la misma velocidad si éste debe incorporar co-procesador matemático. Sin duda, la aparición del 80586 tendrá como efecto una nueva bajada en los precios de los 80486, lo que supondrá la progresiva desaparición de los 80386 (de hecho hoy resulta difícil encontrar un fabricante importante que ofrezca un PC basado en un 80386 con una velocidad por debajo de 33Mhz. Lo mínimo es empezar por los 386SX como configuraciones más baratas, con velocidades de reloj a partir de 20 Mhz, saltar luego al 386 a 33 o 40 Mhz y pasar luego a los 486 empezando por 25Mhz).

Hasta aquí un poco la historia de los PCs, sin pretender sentar cátedra ni ser excesivamente riguroso. No he querido tampoco marear a base de datos como a cuántos MIPS son capaces de correr cada uno de los procesadores a determinada frecuencia de reloj, etc. Pero, ¿por qué he empezado el título diciendo "no es oro todo lo que reluce"?

Aparentemente, viendo la evolución de los uP sería lógico pensar que la velocidad de ejecución de los PC debería ser directamente proporcional a la frecuencia del reloj y a la anchura de los buses (un uP con un bus de datos de 32 bits a la misma frecuencia de reloj debe ser el doble de rápido que un uP con un bus de 16 bits). Sin embargo esto no es del todo cierto. ¿Cuál es la razón?

La razón hay que buscarla en la arquitectura abierta del PC. El PC original tenía cinco ranuras ("slots"). Estas ranuras eran, naturalmente de 8 bits. De las cinco, tres estaban ocupadas (puertos serie y paralelo, tarjeta de vídeo y controlador de disco) y dos quedaban libres.

Cuando se lanzó el primer AT a 6Mhz se modificó el bus para que pudiera correr a una velocidad máxima de 8Mhz. Cuando empezaron a aparecer clónicos que funcionaban a mayores velocidades, los fabricantes, en aras de la compatibilidad, mantuvieron este límite de 8Mhz. A este bus se le llama bus ISA.

Entre tanto, en 1987 IBM lanzó su gama PS, junto con su nuevo sistema operativo OS/2, mucho más avanzado que el MS-DOS, que incluye la multitarea, pero que hasta ahora no ha tenido mucho éxito, ya que requiere mucha memoria (4 Mo mínimo de RAM) y un procesador potente, lo cual no ha sido accesible al gran público hasta hace un año. Además de este operativo, con el fin de cortar las alas a los fabricantes de clónicos, que venden muchos más PCs que IBM, se incluyó en esta gama un bus de diseño propio y exclusivo, del cual en principio no se daba licencia de uso a otras compañías.

Este es un bus de 32 bits de altas prestaciones, pero que ha sido ignorado por los fabricantes de clónicos. De esta manera, la mayoria de PCs que hay en el mercado tienen un verdadero cuello de botella cuando se tienen que comunicar con las tarjetas de expansión, que sólo pueden comunicarse con la CPU a 8 Mhz. Teniendo en cuenta que en las placas bases de la mayoría de los PCs sólo se encuentra la CPU, la memoria y poco más, os podréis dar cuenta del inconveniente.

Como solución, se desarrolló el bus EISA, versión mejorada del ISA. Sin embargo, y por compatibilidad con el bus ISA, sólo puede trabajar a su máxima velocidad, 33 Mhz, cuando se produce una comunicación entre dos tarjetas EISA o entre la

tarjeta EISA y la memoria principal. En el resto de los casos la comunicación se produce a 8Mhz. Además, las tarjetas y los ordenadores con bus EISA son más caros y sus prestaciones se están quedando cortas ante los nuevos y mas rapidos procesadores.

La alternativa MicroChannel (MCA) es aun más cara y creo que incluso IBM ha abandonado este bus en algunos modelos de la gama PS, incluyendo el ISA o al menos ha mezclado el MCA con el ISA. En algunos casos hay fabricantes han desarrollado versiones del bus propias, para aplicaciones que requieran grandes velocidades de transferencia, como tarjetas gráficas o discos duros de gran capacidad.

Otro problema con el que se estan encontrando los ordenadores personales actuales (PCs y resto) es que resulta difícil y caro fabricar tarjetas madres que puedan trabajar a más de 33 MHz. Parece, por tanto, que estamos cerca del límite de velocidad de reloj, lo que no quiere decir que se ha llegado al límite de la velocidad de proceso. Para conseguir velocidades mayores se recurrirá a microprocesadores de tecnología RISC (de juego de instrucciones reducido), se utilizarán microprocesadores de 64 ó más bits, y se utilizarán cada vez más co-procesadores específicos o se integrarán en un mismo chip. Sin embargo, es probable que los diseñadores de chips nos sorprendan con nuevas soluciones cada vez mejores y más potentes. En conclusión, no os dejéis llevar por la publicidad de los PCs. La velocidad del reloj no lo es todo. Tened en cuenta que la mayoría de los programas para PC usan mucho del disco duro, así que un disco duro de acceso lento puede ralentizar aun más las cosas.

En mi experiencia en el trabajo, donde tenemos un AT a 10 Mhz, un PS1 a 12 Mhz y un AT a 33Mhz con un 80486, creo sinceramente que los ordenadores equipados con un uP de la familia 68000 son un 20 ó 25% mas rápidos que un PC con microprocesador de similares características a una misma velocidad de reloj. Esto es una apreciación subjetiva y depende mucho del tipo de programa utilizado (un programa malo puede hacer lento el ordenador más rápido del mundo). Por ejemplo, mi Atari Mega ST2 que tiene un 68000 a 8 Mhz es, para la mayoría de las aplicaciones normales (procesador de texto, hoja de calculo, etc) tan rapido o más que el AT con un 80286 a 10 Mhz, con disco duro de 40 Mo. Utilizar el Windows con el AT 80286 o el PS1 es bastante rollo, ya que usa mucho del disco duro y no soporta la multitarea, siendo bastante lento.

Si tenéis un QL con una Gold Card o algun sistema de emulación en otra máquina mi consejo es que no os paséis a un PC a menos que realmente os sea imprescindible u os podáis comprar un PC con un 80386SX a 25 Mhz como mínimo, aunque hay que reconocer que con los precios tan bajos que hay y lo que estan bajando, a cualquiera le entra la tentación de comprarse uno, aunque sea "como segundo ordenador" (¡toma ya chulería!). Yo estoy estudiando comprarme uno en plan bricolage, comprándomelo por partes (placa madre, tarjetas, caja, etc.) y montándomelo yo mismo, pero no sé dónde lo iba meter y si realmente merece la pena hacerlo.

Espero que este artículo os haya aclarado algunos aspectos de los PCs que no suelen comentarse mucho.

Javier ZUBIETA, 1992 10 12

Novedades QL-Atari

Acabo de recibir el ultimo folleto de Jochen Merz, correspondiente a septiembre. La novedad más importante es la aparición (después de un año de retraso sobre lo inicialmente previsto) de la tarjeta emuladora QL sobre bus VME para el Atari

Mega STE (y para el Atari TT, aunque no está terminado el "software" para este último).

Esta tarjeta no es válida para los modelos ST, STF, STFM, Mega ST o STE, sólo es válida para los Mega STE. Estos modelos vienen alojados en una caja con espacio para un disco duro, teclado separado y disponen de una ranura de expansión tipo bus VME. Estan equipados de un procesador 68000 a 16 MHz y una memoria caché de 16 Ko (no es un sistema de 16 Mhz reales, pero es un 70% más rápido que los ST estándar a 8 Mhz). En los primeros modelos se incluía una disquetera de 3 1/2" de doble densidad (720 Ko), pero desde hace un año vienen como estándar con una unidad de alta densidad (1.44 Mo). La versión más normal es la Mega STE2, con dos megas, interfaz SCSI para disco duro y disco duro incorporado de 48 Mo, pudiéndose sustituir por otro del mismo tipo de mayor capacidad. En algunos países de vende la versión Mega STE1, con 1 Mo de RAM, pero sin la interfaz SCSI ni disco duro. Según los casos, se incluye en la configuración o no, un co-procesador matemático 68882. Pueden tener hasta 4 Mo de RAM, ampliables con modulos SIMM, como la familia STE. El resto de las características son las mismas que las de los modelos STE.

Esta nueva tarjeta ofrece como ventaja más destacada que tiene una resolución gráfica programable que va desde 512x256 pixeles en 4 colores (no se dispone del modo 8) hasta 1024x780 pixeles, dependiendo de la capacidad del monitor, en incrementos de 8 ó 16 pixeles. También se puede programar la frecuencia horizontal hasta 80 Hz (cuanto mayor sea la frecuencia horizontal, la imagen es más estable, eliminándose el parpadeo de la imagen).

Para la instalación no se requiere ningún tipo de soldadura o manipulación del sistema. Sólo hay que insertar la tarjeta en la ranura de expansión.

Viendo la foto que acompaña el folleto, parece que esta tarjeta debe de incorporar un chip de vídeo sencillito (de 40 pines) y una cierta cantidad de circuitería lógica estándar, sin que aparezca la fatídica ULA ZX8301.

Como precio de oferta hasta el día 22 de octubre, Jochen Merz nos ofrece esta

tarjeta a 637 DM = 44.590 pts, pasando posteriormente a 699 DM = 48.930 pts (aproximadamente, teniendo las últimas convulsiones del SME). Como consecuencia de la aparición de este nuevo modelo, la versión anterior del QL-EMULATOR Extended 4 (emulador para los modelos ST, con un modo 4 extendido de 768x280 pixelees) queda con un precio de 377 DM=26.390, hasta que le dure el almacén.

Me parece una opción interesante, si bien algo cara. ¿Veremos alguna vez algo similar para un PC? (De momento ya hay una tarjeta para PC para transformar un PC en un Atari ST. Necesita un PC con un 80386 o superior y parece ser bastante compatible con las aplicaciones escritas siguiendo la reglas de programación de Atari).

En cuanto a los Atari Mega STE2, se pueden encontrar en España a unas 105.000 pts sin disco duro y 156.000 pts con disco duro de 48 Mo (sin monitor. Se puede utilizar un televisor estándar o un televisor con euroconector). En Alemania un Mega STE 1 con disco duro de 48 Mo vale unos 1.400 DM = 98.000 Pts, y en Inglaterra un Mega STE2 con disco de 48 Mo cuesta unas 700 = 126.000 Pts.

Programación en SuperBASIC orientada al objeto

Para usar un objeto basta unir su fichero con el del programa que se esté realizando y escribir al principio la instrucción Obj_Arranca, donde "Obj" será el identificador del objeto

En general, es aconsejable arrancar primero los objetos que van a necesitar otros objetos. Es decir, seguir el orden expuesto:

Objeto	Identificador	Objetos que necesita
Ventana	Ven	
Sonido	Son	
Usuario	Usr	Ventana, Sonido
Fichero	Fch	Usuario
Tiempo	Tim	
Tecla	Tec	
Cadena	Cad	

```
REMark *-------------------------------------------------------------------
REMark * FICHERO:  Ventana_blq
REMark * OBJETIVO: Definición del objeto "Ventana" (Ven)
REMark * FECHA:    D.27.9.1992
REMark * FUNCIONES O PROCEDIMIENTOS DISPONIBLES:
REMark *   Ven_Arranca    Ven_Crea%()    Ven_Limpia      Ven_Borde
REMark *   Ven_Cursor     Ven_Papel      Ven_Tinta       Ven_TamCar
REMark *   Ven_Escribe    Ven_Cierra
REMark *-------------------------------------------------------------------
:
REMark *-------------------------------------------------------------------
REMark * FICHERO:  Sonido_blq
REMark * OBJETIVO: Definir el objeto "Sonido" (Son)
REMark * FECHA:    J.29.10.1992
REMark * OBJETOS UTILIZADOS:
REMark * FUNCIONES O PROCEDIMIENTOS DISPONIBLES:
REMark *   Son_Arranca   Son_Cambia   Son_Error
REMark *-------------------------------------------------------------------
:
REMark *-------------------------------------------------------------------
REMark * FICHERO:  Usuario_blq
REMark * OBJETIVO: Definir el objeto "Usuario" (Usr)
REMark * FECHA:    D.1.11.1992
REMark * OBJETOS UTILIZADOS: Ventana, Sonido
REMark * FUNCIONES O PROCEDIMIENTOS DISPONIBLES:
REMark *   Usr_Arranca       Usr_EnterEsc       Usr_Avisa    Usr_AvisaError
REMark *   Usr_PideTexto$()  Usr_PideNatural%()  Usr_Informa
REMark *-------------------------------------------------------------------
:
```

```
REMark *------------------------------------------------------------------
REMark * FICHERO:  Fichero_blq
REMark * OBJETIVO: Definir el objeto "Fichero" (Fch)
REMark * FECHA:    J.29.10.1992
REMark * OBJETOS UTILIZADOS: Usuario
REMark * FUNCIONES O PROCEDIMIENTOS DISPONIBLES:
REMark *   Fch_Arranca   Fch_Abre%()   Fch_Cierra   Fch_Lee$()   Fch_Escribe
REMark *------------------------------------------------------------------
:

REMark *------------------------------------------------------------------
REMark * FICHERO:  Tiempo_blq
REMark * OBJETIVO: Definir el objeto "Tiempo" (Tim)
REMark * FECHA:    V.30.10.1992
REMark * FUNCIONES O PROCEDIMIENTOS DISPONIBLES:
REMark *   Tim_Arranca  Tim_Fecha$()  Tim_Hora$()  Tim_Crono
REMark *------------------------------------------------------------------
:
REMark *------------------------------------------------------------------
REMark * FICHERO:  Tecla_blq
REMark * OBJETIVO: Definir el objeto "Tecla" (Tec)
REMark * FECHA:    V.30.10.1992
REMark * FUNCIONES O PROCEDIMIENTOS DISPONIBLES:
REMark *   Tec_Arranca   Tec_Pulsada
REMark *------------------------------------------------------------------
:
REMark *------------------------------------------------------------------
REMark * FICHERO:  Cadena_blq
REMark * OBJETIVO: Definir el objeto "Cadena" (Cad)
REMark * FECHA:    V.30.10.1992
REMark * FUNCIONES O PROCEDIMIENTOS DISPONIBLES:
REMark *   Cad_OrdenaF   Cad_OrdenaI   Cad_Funde   Cad_Busca%()
REMark *------------------------------------------------------------------
```

Novedades en el mundo Atari: Falcon

Se acaba de presentar un nuevo ordenador Atari. Su nombre es Falcon y está basado en un uP 68030 a 16 Mhz, siendo compatible en sentido ascendente con la familia ST. Dispone de unas prestaciones realmente impresionantes:

- Microprocesador 68030 de 32 bits a 16 Mhz (es una lástima que no hayan optado por una velocidad un poco superior, tal vez 20 o 24 Mhz).
- Posibilidad de incorporar un coprocesador matematico 68882.
- Memoria RAM: 1, 4 ó 14 Mo.
- Memoria ROM: 512 Ko, con el interfaz gráfico GEM incorporado (iconos, ventanas y menúes).
- Unidad de disco de 3 1/2" de 1.44 Mo (lástima también que no hayan incluido una unidad de 2.88 Mo, aunque es fácil que lo hagan en un futuro próximo).
- Disco duro: incorpora un bus IDE para colocar opcionalmente un disco duro interno de 65 Mo.
- Incorpora un procesador de señales digitales DSP56001 a 32Mhz. Se trata de un potentísimo chip de uso general capaz de manipular a altísima velocidad cualquier tipo de señales digitales. Su campo de aplicación es potencialmente amplísimo.
- Gráficos: todos los modos del ATARI STE, más modos VGA.

Modos STE:
  - 320 X 200 pixeles en 16 colores de una paleta de 4096.
  - 640 X 200 pixeles en 4 colores de una paleta de 4096.
  - 640 X 400 pixeles en 2 colores de una paleta de 4096.

Modos VGA:
  - 320 X 200 pixeles en 2, 16 o 256 colores de una paleta de 262.144.
  - 640 X 400 pixeles en 2, 16 o 256 colores de una paleta de 262.144.
  - 640 X 480 en 256 colores de una paleta de 262.144.

Modo de color real ("true colour"): 320 X 200 pixeles en 32.768 colores de una paleta de 262.144.

- Acelerador de gráficos ("blitter") incorporado.

- Sonido: 8 canales en estéreo, con conversión A/D de 16 bits (calidad superior a la del disco compacto).
- Sistema operativo TOS multitarea.
- Conexión DMA tipo SCSI II (estándar, lo que le permite la conexión con dispositivos Mac o PC que tengan esta conexión como, por ejemplo, discos duros o CD-ROM).
- Entrada directa de micrófono, para grabar directamente la voz u otras fuentes sonoras, con posibilidad de efectos en tiempo real.
- Salida DSP (para la conexión de CD-ROM, faxes, etc, si bien no sé si este tipo de conector está estandarizado).
- Conexiones MIDI.
- Bus de expansión interno.
- Puertos serie, paralelo, monitor, etc.

Los diseñadores de Atari se las han ingeniado para meter todas estas maravillas dentro de la misma caja de los Atari ST/STE (teclado y ordenador integrados). El precio de la version básica (1 Mo, sin disco duro) es de 499 (=89.820 Pts.) y de 899 (=161.820 Pts.) con un disco duro interno de 65 Mo. Para el año que viene está previsto un modelo basado alrededor del uP 68040, posiblemente con una paleta de 16 millones de colores y ranuras de expansión VME, con la UCP en una caja y teclado separado.

Con estas especificaciones se supera de largo a la actual gama Atari (excepto el modelo TT que tiene un 68030 a 32Mhz, aunque no dispone de un sistema de sonido ni de gráficos tan avanzado ni, por supuesto, del DSP), siendo muy superior al Commodore Amiga.

Hay rumores de que Commodore está ya trabajando en una respuesta conveniente, que posiblemente incorpore el chip DSP56001, auténtica maravilla que puede marcar la diferencia entre estos nuevos ordenadores y los precedentes, incluso los PCs.

La potencia de este chip es tan grande que permitiría, por ejemplo, conectar el Falcon a una cadena de alta fidelidad y, mediante el programa adecuado, actuar como un ecualizador en tiempo real, con un sonido de disco compacto.

Con este tremendo potencial, el implementar un emulador de QL en "software" para esta nueva máquina estaría "chupado". A ver si se anima alguien y lo pone en el dominio público. De esta manera se obtendría un QL unas 8 veces más rapido. Por otro lado, si, como es previsible, se lanza un nuevo modelo con bus VME, se podría utilizar la nueva placa emuladora de Jochen Merz, con lo cual podríamos tener QL garantizado por bastantes años.

Potencialmente, este ordenador es superior a cualquier PC basado en un uP 80386SX, y similar a uno basado en un 80386DX a 16-20 Mhz, a un precio muy interesante, aunque los PCs estan bajando continuamente de precio.

Si Atari puede levantar el vuelo (nunca mejor dicho) con este nuevo ordenador, ante la presión de los PCs, estoy decidido a comprarme uno en cuanto saquen un modelo con teclado separado y ranuras VME de expansión.

Atari ha corregido alguno de sus fallos anteriores (todos los fabricantes los tienen, no sólo el tío Clive o el amigo Sugar) al optar por unos modos gráficos estándar (VGA) y un puerto DMA SCSI II estándar, en lugar del anterior de uso propio y exclusivo, que encarecía el precio de los discos duros y no permitía la conexión sencilla de dispositivos tales como escaneadores en color, etc.

A pesar de lo mal que le van las cosas a Atari (y, por cierto, lo mal que se están poniendo para los Commodore Amiga ante el empuje PC) estoy seguro de que habrá apoyo para el ST durante bastante tiempo, aunque pasado mañana Atari tuviera que cerrar sus puertas. Su gran base de usuarios (mas de 2.5 millones de unidades vendidas) y, sobre todo, que hayan visto la luz desarrollos posteriores al ST básico (diseño del año 1985), lo cual no ha sucedido, por desgracia, en el QL, me dan suficiente confianza.

Si finalmente sucumbo ante los PCs lo haré resignadamente, pero no creo que me rasge las vestiduras. Sin embargo, creo que ese momento tardará en producirse.

Javier ZUBIETA, 1992 10 12

Coherent

Lexicon -- Introduction

The Mark Williams Lexicon is a new approach to documentation of computer software. The Lexicon is designed to improve documentation and eliminate some limitations found in more conventional documentation.

How to Use the Lexicon
The Lexicon consists of one large document that contains entries for every aspect of COHERENT. You will not have to search through a number of different manuals to find the entry you are looking for.

Every entry in the Lexicon has the same structure. The first line gives the name of the topic being discussed, followed by its type (e.g., Command).

The next lines briefly describe the item, then give the item's usage, where applicable. These are followed by a brief discussion of the item, and an example.

Cross-references follow. These can be to other entries or to other texts. Diagnostics and notes, where applicable, conclude each entry.

Internally, the Lexicon has a tree structure. The ''root'' entry is the present entry, for Lexicon. Below this entry comes the set of Overview entries. Each Overview entry introduces a group of entries; for example, the Overview entry for libraries introduces all of the libraries that come with COHERENT. An Overview entry points to other entries, some of which may themselves be Overview entries that point to still other entries. For example, under libraries come the Overview entries string and system calls, which point, respectively, to all string functions available in COHERENT's C libraries and to all of COHERENT's system calls.

Each entry cross-references other entries. These cross-references point up the documentation tree, toward the entry's ''parent'' entry (usually an Overview entry); own the tree to subordinate entries; and across to entries on related subjects. For example, the entry for getchar cross-references STDIO, which is its Overview article, plus putchar and getc, which are related entries of interest to the user. STDIO, in turn, points up to the article on libraries, which in turn points to the root article, Lexicon. The Lexicon is designed so you can trace from any one entry to any other simply by following the chain of cross-references up and down the documentation tree.

Use the Lexicon
If, while reading an entry, you encounter a technical term that you do not understand, look it up in the Lexicon. You should find an entry for it. For example, if a function is said to return a data type float and you do not know exactly what a float is, look it up. You will find it described in full. In this way, you should increase your understanding of COHERENT, and make your programming easier and more productive.

Overview Articles
The Lexicon includes the following overview articles. Look at the appropriate overview article for information on the subject in which you are interested. The overview article will give you an overview of the topic, and tell you which Lexicon articles you should read to find detailed information.

C language
    This article summarizes COHERENT's implementation of the C language. It introduces subordinate articles, such as those that describe each C keyword.

commands
    This article briefly summarizes each COHERENT system command.

definitions
    The Lexicon includes a number of articles that define technical terms that are used through it. This overview article lists the definition

articles included in the Lexicon.

device drivers
This article introduces COHERENT's suite of device drivers, and points to subordinate articles that describe each driver in details.

environmental variables
This article lists the commonly used environmental variables that are described in the Lexicon.

file formats
The COHERENT system has a number of special files that contain information presented in a special format. Some files are meant to be read mechanically, such as executable files or relocatable objects; others you can edit to change the behavior of one or another COHERENT system. This overview article introduces the subordinate articles that describe the formats of these special files.

libraries
This introduces the libraries included with the COHERENT system, for use with the COHERENT C compiler, and the families of functions in each.

system maintenance
Certain files and commands are used only to help you maintain your COHERENT system and help it run smoothly. This article introduces the subordinate articles that describe the COHERENT system's tools for to help you perform system maintenance.

technical information
Finally, the Lexicon contains a set of articles that do not easily fit into any other category. These give broad technical information, both to help you decypher other articles within the Lexicon, and to provide you with a ''cookbook'' with which you can solve common problems. The article names should be self-explanatory, e.g., terminal, printer, and RS-232. If you're trying to tackle a new problem and don't have a clue as to where to begin, check this overview article first. You may well find that it lists a subordinate article that is helpful.

The following lists the commands included with COHERENT. The command name is given on the left and a description on the right.

Communications
The following commands let you exchange information with other users and other systems.

```
ckermit........Interactive inter-system communication and file transfer
kermit.........Communication and file transfer (COHERENT 286 only)
mail...........Send/read electronic mail
mesg...........Permit/deny messages from other users
msg............Send a brief message to other users
msgs...........Read messages intended for all COHERENT users
uucico.........Connect to a remote system
uucp...........Copy a file to or from a remote system
wall...........Send a message to all logged in users
write..........Converse with another user
```

Device Handling
The following commands help you run peripheral devices, especially printers. For commands that drive communications devices, e.g., modems, see the section on Communications, above.

```
epson..........Print a file on an Epson printer
fnkey..........Set/print function keys for the console
hp.............Prepare files for HP LaserJet-compatible printer
hpr............Send to LaserJet printer spooler
hpskip.........Abort/restart current listing on LaserJet
lpr............Send to line printer spooler
lpskip.........Terminate/restart current line printer listing
stty...........Set/print terminal modes
```

tty............Print the user's terminal name
ttystat........Get terminal status

Directory and File Handling
The following commands let you create, remove, and otherwise manipulate files and directories.

cat............Concatenate/print files
cd.............Change directory
chgrp..........Change the group owner of a file
chmod..........Change the modes of a file
chmog..........Change mode, ownership, and group of a file
chown..........Change ownership of a file
cmp............Compare bytes of two files
compress.......Compress a file
cp.............Copy a file
cpdir..........Copy directory hierarchy
dd.............File conversion
dos............Manipulate files on MS-DOS file systems
doscat.........Concatenate a file on an MS-DOS file system
doscp..........Copy files to/from an MS-DOS file system
doscpdir.......Copy directories to/from an MS-DOS file system
dosdir.........List the contents of an MS-DOS directory
dosdel.........Delete a file from an MS-DOS file system
dosformat......Build MS-DOS file system on a floppy disk
doslabel.......Label an MS-DOS floppy disk
dosls..........List files on an MS-DOS file system
dosmkdir.......Create a directory in an MS-DOS file system
dosrm..........Remove a file from an MS-DOS file system
dosrmdir.......Remove a directory from an MS-DOS file system
fdisk..........View/change hard-disk partitioning
file...........Name a file's type
find...........Search for files satisfying a pattern
l..............List directory's contents in long format
lc.............List directory's contents in columnar format
lf.............List directory's contents in columnar format
ln.............Create a link to a file
lr.............List subdirectorys' contents in columnar format
ls.............List directory's contents
lx.............List directory's contents in columnar format
mkdir..........Create a directory
mv.............Rename files or directories
mvdir..........Rename a directory (COHERENT 386 only)
pwd............Print the name of the current directory
qfind..........Quickly find all files with a given name
rm.............Remove files
rmdir..........Remove directories
touch..........Update modification time of a file
uncompress.....Uncompress a file
whereis........Locate source, binary, and manual files
which..........Locate executable files
zcat...........Concatenate a compressed file

Editors
COHERENT includes a number of text editors, to suit a variety of tastes.

ed.............Interactive line editor
elvis..........Berkeley-style screen editor
ex.............Berkeley-style line editor
me.............MicroEMACS screen editor
sed............Stream editor
vi.............Berkeley-style screen editor

Games
The following commands are just for fun.

banner.........Print large sized letters
cal............Print a calendar
chase..........Highly amusing video game
fortune........Print randomly selected, hopefully humorous, text

guess..........Extraordinarily amusing guessing game
lines..........Highly amusing board game
moo............Greatly amusing numeric guessing game
rubik..........Play Rubik's cube
ttt............Three-dimensional tic-tac-toe

Languages and Programming Tools
The COHERENT system comes with a number of languages, and tools for debugging and maintaining your programs.

as.............Mark Williams assembler
asfix..........Convert file to 80386 as form (COHERENT 386 only)
awk............Report generation, pattern scanning, and processing language
cc.............C-language compiler
cdmp...........Dump COFF files into a readable form (COHERENT 386 only)
conv...........Numeric base converter
cpp............C preprocessor
db.............Assembler-level symbolic debugger
fixstack.......Alter size of a program's stack (COHERENT 286 only)
ld.............Link relocatable object files
lex............Lexical analyzer generator
m4.............Macro processor
make...........Program building discipline
nm.............Print a program's symbol table
od.............Print an octal dump of a file
prof...........Print execution profile of a C program
ref............Display a C function header
srcpath........Find source files
size...........Print size of an object file
strip..........Strip symbol tables from executable file
yacc...........Parser generator

Libraries and Archives
The following commands help you create and read libraries and archives. These can be used as libraries (such as the libraries used when linking a C program), or to back up files.

ar.............The object librarian/archiver
cpio...........Archiving/backup utility
dump...........File-system backup utility
dumpdate.......Print dump dates
dumpdir........Print the directory of a dump
pax............Portable archive interchange
ranlib.........Create index for object library
restor.........Restore file system
tar............V7 tape archive manager
ustar..........Tape archive utility

Mail
COHERENT comes with with a full-featured, UNIX-style mail facility. This is described in the overview article mail. The following commands perform mail-related work. Note that some are also listed in other sections of this article. Note, too, that the descriptions for smail and rmail are only for those users who wish to manipulate UUCP mailing on a low level; for most users, the descriptions under the command mail are more than sufficient.

cvmail.........Convert stored mail to System V format
lmail..........Deliver local mail
mail...........Send/read electronic mail
mkfnames.......Generate data base of user names
nptx...........Generate permutations of users' full names
rmail..........Receive UUCP mail
smail..........Send UUCP mail

Shell Commands
COHERENT comes with two command interpreters, or shells: ksh, the Korn shell, and sh, the Bourne shell. The following commands are used either by the Korn shell, by the Bourne shell, or by both. Please note that commands used only by the Korn shell are marked by a dagger '+', whereas commands used only by the Bourne shell are marked by an asterisk '*'.

```
alias+.........Set an alias
basename.......Strip path information from a file name
bind+..........Bind key sequence to editing command
break..........Exit from shell construct
builtin+.......Execute a command as a built-in command
case...........Execute commands conditionally according to pattern
cd.............Change directory
continue.......Terminate current iteration of shell construct
dirs*..........Print contents of directory stack
echo...........Repeat an argument
eval...........Evaluate arguments
exec...........Execute command directly
exit...........Exit from a shell
export.........Add a shell variable to the environment
expr...........Compute a command line expression
false..........Unconditional failure
fc+............Edit and re-execute one or more previous commands
for............Execute commands for tokens in list
from...........Generate list of numbers, for use in loop
getopts+.......Parse command-line options
hash+..........Add a command to the shell's hash table
if.............Execute a command conditionally
jobs+..........Print information about jobs
let............Evaluate an expression
nohup..........Run a command while ignoring hangup signals
popd*..........Pop an item from the directory stack
prep...........Produce a word list
print+.........Echo text onto the standard output
pushd*.........Push an item onto the directory stack
read...........Assign values to shell variables
readonly.......Mark a shell variable as read only
set............Set shell option flags and positional parameters
shift..........Shift positional parameters
sleep..........Stop executing for a specified time
tee............Branch pipe output
test...........Evaluate conditional expression
times..........Print total user and system times
trap...........Execute command on receipt of signal
true...........Unconditional success
typeset+.......Set/list variables and their attributes
umask..........Set the file-creation mask
unalias+.......Remove an alias
until..........Execute commands repeatedly
wait...........Await completion of background process
whence+........List a command's type
while..........Execute commands repeatedly
```

String Processing

Some of the most useful commands are those that process strings. COHERENT has many commands that search for strings, manipulate strings, sort strings, and otherwise perform useful manipulations on strings.

```
c..............Print multi-column output
cgrep..........Pattern search for C programs
comm...........Print common lines
cut............Select portions of each line of a file
detab..........Replace tab characters with spaces
diff...........Summarize differences between two files
diff3..........Summarize differences among three files
egrep..........Extended pattern search
grep...........Pattern search
head...........Print the beginning of a file
join...........Join two data bases
look...........Find matching lines in a sorted file
more...........Display text one screenful at a time
paste..........Merge lines of files
rev............Print text backwards
scat...........Print text files one screenful at a time
sort...........Sort lines of text
```

```
split..........Split a text file into smaller files
strings........Print all character strings from a file
tail...........Print the end of a file
tr.............Translate characters
tsort..........Topological sort
uniq...........Remove/count repeated lines in a sorted file
view...........Berkeley-style text viewer
wc.............Count words, lines, and characters in text files
```

System Accounting
The following commands help you to keep track of how your COHERENT system is working.

```
ac.............Summarize login accounting information
accton.........Enable/disable process accounting
df.............Measure free space on disk
du.............Summarize disk usage
ps.............Print process status
sa.............Print a summary of process accounting
quot...........Summarize file-system usage
time...........Time the execution of a command
times..........Print total user and system times
uulog..........Examine UUCP operations
```

System Maintenance
These commands help you to maintain your COHERENT system.

```
at.............Execute commands at given time
bad............Maintain list of bad blocks
badscan........Examine a device for bad blocks
build..........Install COHERENT onto a hard disk
check..........Check file system
clri...........Clear i-node
crontab........Copy a command file into the crontab directory
date...........Print/set the date and time
dcheck.........Check directory consistency
drvld..........Load loadable drivers into memory (COHERENT 286 only)
fdformat.......Low-level format a floppy disk
fsck...........Check and repair file systems interactively
icheck.........i-node consistency check
man............Print Lexicon entries
mkfs...........Make a new file system
mknod..........Make a special file or named pipe
mount..........Mount a file system
ncheck.........Print file names corresponding to i-node
newgrp.........Change to a new group
newusr.........Add new user to COHERENT system
reboot.........Reboot the COHERENT system
shutdown.......Shut down the COHERENT system
sync...........Flush system buffers
umount.........Unmount a file system
unmkfs.........Create a prototype file system
uucheck........Sanity-check the UUCP system
```

terminfo
COHERENT 386 supports an implementation of terminfo, the terminal- description utility used under UNIX System V. (It also supports termcap, should you prefer to use that venerable, but still useful, system.) The following commands help support terminfo:

```
captoinfo......Convert termcap data to terminfo form (COHERENT 386 only)
infocmp........De-compile a terminfo binary file (COHERENT 386 only)
tic............Compile a terminfo description (COHERENT 386 only)
```

Text Processors
These commands help you to create orderly, attractive printed text. For information on how to print the output of these commands, see the commands listed under Device Handling, above.

```
col............Remove reverse and half line motions
deroff.........Remove text formatting control information
```

nroff..........Text-formatting language
fwtable........Build a font-width table from PCL or PostScript font
pr.............Paginate and print files
prps...........Paginate and print files on PostScript printers
spell..........Find spelling errors
troff..........Extended text-formatting language
typo...........Detect possible typographical and spelling errors

UUCP
The UUCP commands lets you form a network with other COHERENT or UNIX systems. Members of the network can grant each other permission to exchange mail and execute commands on each others' systems remotely and automatically, without having to be directed by a human being. The overview article UUCP describes the COHERENT UUCP facility in some detail. The following commands perform UUCP-related work; note that some of the commands listed here also are also listed in other sections of this article.

uucheck........Sanity-check the UUCP system
uucico.........Connect to a remote system
uucp...........Copy a file to or from a remote system
uudecode.......Decode a transmitted UUCP file
uuencode.......Encode a UUCP file for tranmission
uuinstall......Configure UUCP control files
uulog..........Examine UUCP operations
uumvlog........Archive UUCP log files
uuname.........Print names of recognized systems
uurmlock.......Remove UUCP lock files
uutouch........Force polling of a remote site
uux............Execute a command on a remote system
uuxqt..........Execute file as requested by remote system

Miscellaneous
The following commands do not fit neatly into any of the above categories. These include some of the more interesting and useful COHERENT commands, and are worth your attention.

ATclock........Read/set the AT realtime clock
bc.............Interactive calculator with arbitrary precision
calendar.......Electronic reminder service
chroot.........Change root directory
clear..........Clear your terminal's screen
coh_intro......Tour the COHERENT file system
crypt..........Encrypt/decrypt text
dc.............Desk calculator
disable........Disable a port
enable.........Enable a port
env............Execute a command in an environment
factor.........Factor a number
help...........Print concise description of command
install........Install a software update onto COHERENT
kill...........Signal a process
ksh............Invoke the Korn shell
login..........Log in or change user name
passwd.........Set/change login password
phone..........Print numbers and addresses from phone directory
script.........Capture a terminal session into a file
sh.............Invoke the Bourne shell
su.............Substitute user id, become superuser
sum............Print checksum of a file
uname..........Print information about the system
units..........Convert measurements
who............Print who is logged in
yes............Print infinitely many responses

Creta: un ejemplo de POO

Indice

1. Generalidades sobre la programación orientada al objeto (POO)
2. Origen de CRETA
3. Implementación de CRETA
4. Adaptación al QL
5. Uso de CRETA
6. El programa de demostración
7. Nuevos laberintos

1. Generalidades sobre la programación orientada al objeto (POO)

En la breve historia de la informática se pueden distinguir varias etapas: Los años 50 y 60 se encuentran dominados por los desarrollos de "hardware", los años 70 representan una etapa de transición, y a partir de los 80 el "software" se convierte en el núcleo de los desarrollos informáticos, queriendo esto decir que la mayor parte de los recursos económicos y humanos de un proyecto informático se gastan en desarrollar el "software" del sistema.

Cuando crece el volumen de "software" que hay que desarrollar, se encuentran problemas:

* El "software" es mucho menos fiable que el "hardware".
* Los plazos de desarrollo del "software" se hacen muy difíciles de cumplir.
* El "software" es muy difícil de mantener.

La programación estructurada comienza a dar respuesta a estos problemas. Se intenta modularizar el "software" y a aplicar estrictamente criterios de calidad y fiabilidad en cada fase de desarrollo.

La POO es un paso más sobre estas ideas. Permite que el análisis, diseño e implementación de un proyecto estén siempre orientados al mundo real, más que orientados a la máquina, lo que facilita la continuidad de estas fases.

La POO permite encapsular y reutilizar "software" de un modo mucho más eficiente que con otros métodos. Los objetos son componentes de "software" mucho más fiables que los módulos. Su mantenimiento es mucho más sencillo.

Una aplicación orientada al objeto se organiza alrededor de entidades llamadas "objetos", que se comunican por medio de "mensajes". Como veis, como en el mundo real.

Cuando hacemos análisis orientado al objeto debemos identificar objetos y asignar responsabilidades a cada uno de ellos. En las fases de diseño e implementación se definen las estructuras necesarias para albergar la información y se programan las funciones que les permiten cumplir las responsabilidades que les hemos asignado. Es decir, en los objetos reunimos datos (estructuras) y procedimientos para manejarlos (métodos).

Cuando miramos un objeto "desde fuera", es decir, cuando lo usamos en una aplicación, sólo vemos de ellos un interfaz en el que se especifica qué cosas sabe hacer el objeto y cómo se han de pedir; por tanto, vemos los mensajes que podemos mandar al objeto. Debe quedar claro que esto permite separar completamente la implementación de un objeto de su utilización: el responsable de un objeto podría cambiar por completo la estructura y los métodos, pero si no cambia el interfaz de los mensajes, todos los cambios resultarán transparentes a los programadores que utilicen ese objeto.

Resumiendo, para desarrollar una aplicación orientada al objeto podemos seguir estas fases:

I) Análisis
   * Identificar objetos
   * Asignar responsabilidades
II) Diseño
   * Especificar cada objeto
     . Identificar nuevos objetos
     . Definir estructura: pública y privada
     . Especificar métodos: públicos (interfaz) y privados
III) Implementación
   * Definir estructuras
   * Programar métodos

Una advertencia: si vamos a hacer un pequeño programa, todo esto normalmente nos va a dar más trabajo que si hicieramos el programa "a pedal"; cuanto mayor es el programa y más programas hagamos, más provecho sacaremos a esta filosofía de trabajo. En todo caso, hay muchas ideas que podemos aplicar a nuestros desarrollos y hacelos más claros y productivos aunque no se siga estrictamente la metodología de orientación al objeto.

Tengo que apuntar que la POO tiene en la "herencia" el concepto más potente, y uno de los que más tiempo ahorra en grandes desarrollos.

2. Origen de Creta

En el curso "Sistemas expertos" organizado por la CAM en 1991-92 teníamos la asignatura "C", durante la cual se nos presentaron los conceptos elementales de la POO, y como ejemplo práctico se utilizó el siguiente problema: Programar un robot que sea capaz de salir de un laberinto.

Los profesores de la asignatura, pertenecientes a la empresa "EQ Sistemas inteligentes", analizaron el problema, diseñaron el programa y por fin lo implementaron. Las dos primeras fases sirvieron de soporte a las explicaciones, pero la tercera ya no fue explicada en clase.

Durante el análisis aparecen tres objetos: El laberinto, el robot y el paso. Este tercer objeto es "interno": es un objeto que necesita el robot.

3. Implementación de Creta

El principal responsable de esta fase fue Antonio Salmerón Cabañas. El resultado de su trabajo fueron estos ficheros:

Robot.h -> Definición del objeto robot y de sus métodos elementales; declaración del resto de los métodos.
Robot.c -> Programación de los métodos del robot.
Laberint.h, Laberint.c, Paso.h, Paso.c -> Análogamente para los objetos laberinto y paso.
Ifz.h -> El interfaz de pantalla se considera un objeto y se define mediante macros.
Vrobot.c -> La aplicación principal.

Para facilitarme el manejo de la aplicación yo uní todos los ficheros en uno sólo, que llamé Creta.c. Programé la posibilidad de mandar distintos parámetros al programa por medio de la línea de comandos.

4. Adaptación al QL

De esta parte me ocupé yo, pero ha sido muy sencillo, dado que el programa está muy bien hecho. Sólo hizo falta cambiar los macros de manejo del interfaz de pantalla. Para que quedara en poco más majo, definí nuevos caracteres para el muro del laberinto, para el robot (que ahora es una especie de hormiga) y para las huellas que va dejando el robot. Para aprovechar que el QL tiene más columnas que el PC, amplié los límites máximos del laberinto.

"EQ Sistemas inteligentes" ha dado permiso para que el programa sea distribuido libremente entre los socios de QLíper, no estando permitida su utilización por personas ajenas al club. Por tanto, no es un programa de dominio público.

5. Uso de Creta

Es necesario Toolkit II. Se puede lanzar en multitarea, aunque no le veo el sentido. Por tanto lo lógico es usar EW Creta_exe. Ahora bien, lo bonito es usar distintas opciones, aparte de las que toma el programa por defecto. Las opciones se mandan siguiendo la notación del TKII: EW Creta_exe; opciones$

donde opciones$ es una cadena que puede tener cualquiera de estas opciones (en cualquier número, en cualquier orden y en mayúsculas o minúsculas):

-l : Para poner el nombre del laberinto que se desea cargar.
-f : Para indicar en qué fila se debe situar el robot (la de arriba es la 0).
-c : Para indicar en qué columna se debe situar el robot (la de la izquierda es la 0).

-d : Para colocar al robot mirando en la dirección deseada.

Así, si escribimos EW Creta_exe; "-lMiLab_lab -dN -F5"

cargaremos el laberinto MiLab_lab, pondremos el robot mirando al norte y en la fila 5. Como no se indica la columna, el programa toma por defecto la 1.

El programa tiene dos pantallas. En la primera se presenta el programa y los parámetros elegidos. Algunos errores se indican en esta pantalla. En la segunda se realiza la carga del laberinto y la ubicación del robot en el laberinto. Pueden aparecer mensajes de error, pero si todo ha ido bien, el robot buscará la salida del laberinto.

Primero sigue la estrategia de girar siempre hacia la izquierda; si esa estrategia le lleva a un bucle, cambia de estrategia y va siempre hacia la derecha. Si aún así se encuentra en un bucle, se declara encerrado y se para.

Si el laberinto tiene entrada y salida, es absolutamente seguro que al colocar el robot en la entrada, encontrará la salida siguiendo la primera estrategia.

El único caso en que puede quedar encerrado es cuando se le coloque dentro del laberinto.

Cada vez que el robot da un paso, comprueba en su memoria si es uno de los pasos que ya había dado. Esto lo hace tan deprisa que si no pusiéramos un retardo, no se vería el movimiento del robot. El macro ROB_RETARDO controla eso. Si queréis cambiar la velocidad del robot, hay que cambiar el valor de ese macro y volver a compilar. He preparado dos versiones del ejecutable. Creta_exe para los QL a 7.5 MHz y CretaGC_exe para los de 16MHz.

6. El programa de demostración

El programa Creta_Demo_bas estaba pensado para mostrar una serie de ejemplos distintos. El programa debía ir mostrando en sucesión los distintos casos sin apenas intervención humana. Pero hay un error en C68 que hace que el programa en SB aborte al concluir el ejecutable, de modo que he tenido que reescribir la demostración.

Creta_Demo_bas hace lo siguiente:

* Copia el ejecutable en RAM (los que uséis Gold Card debéis editar esa línea).
* Coloca la pantalla en modo 4.
* Define una serie de ALTKEYs con ejemplos de Creta.
* Recuerda al usuario los ALTKEYs utilizados.

Ahora debéis ir pulsando ALT/1 , ALT/2, etc., para ir viendo los ejemplos.

7. Nuevos laberintos

Quizá a alguno de vosotros os apetezca crear algún laberinto para poner en dificultades el robot. La estructura de los laberintos es muy sencilla, y permite crearlos con cualquier editor de textos.

Un laberinto es un fichero ASCII con estas características:

La primera línea del fichero indica cuántas filas y columnas tiene el laberinto (se admiten 21 filas y 81 columnas como máximo).

Las restantes filas del fichero definen las filas del laberinto. El carácter 32 (el espacio) indica un hueco y cualquier otro carácter significa un muro en el laberinto; yo suelo utilizar "X".

Aconsejo que no pongáis ningún hueco en la columna de la derecha. Creo que el programa no lo leería bien.

El laberinto Marco_lab os puede servir como base para realizar nuevos laberintos. Si alguien realiza alguno, me encantaría tenerlo.

Pedro REINA, 1992 07 12

Los libros del C

No hace mucho tiempo, año 7 después de Sinclair, el único libro que se podía encontrar en España para aprender C era "The C Programming Language" (Kernighan, Brian W., y Ritchie, Dennis M. Prentice-Hall, 1978). Como era natural, en inglés. Yo nunca pude encontrarlo ni comprarlo, pero por lo que se rumoreaba, es un libro que no debería faltar en la biblioteca de un programador en C.

El primer libro que se ha publicado en España sobre C traducido al castellano es "Programación en C. Introducción y conceptos avanzados" de Anaya Multimedia. Yo tengo la primera edición. Ya va por la tercera edición ampliada y mejorada. Esta última edición tiene más de 200 páginas que la primera e incluye discos de 3.5" y 5.25" en formato MS-DOS con los ejemplos del libro (ideal para no teclear). Si alguien desea aprender C, este es su libro y no necesita ninguno más (cuesta unas 7.500 ptas).

"Lenguaje C. Biblioteca de Funciones" (Osborne/Mc Graw-Hill) es un libro plagado de funciones, pero la mayoría podéis encontrarlas ya definidas en una librería como la del C68. Es solamente interesante para ver ejemplos escritos en C. La mayor pega que debo advertir es que en la traducción, el traductor se comió en algunos ejemplos líneas completas de código e incluso tradujo funciones estándar que no debería traducir. En otras palabras, los listados no son volcados de ficheros originales, son traducciones directas del libro inglés original. Algo se aprende, pero en mi opinión, no vale la pena comprarlo.

"ANSI C a su alcance" (Osborne/McGraw-Hill) es un libro que he comprado porque se supone que contiene todo el estándar ANSI C. Aunque parece que no está nada mal, mucho me temo que casi todo su contenido podéis encontrarlo en el libro "Programación C". Podríamos decir que se trata de un libro de la serie B.

"La Biblia del Turbo C" (Anaya Multimedia) es el típico libro que tenemos que comprar todos aquellos que hemos obtenido de alguna manera una copia pirata del Turbo C para MS-DOS. Viene con un par de discos 5.25" con los ejemplos del libro y programas completos. Mi principal interés es escribir mis futuros programas C68 intentando que más tarde las modificaciones para compilarlos con Turbo C sean mínimas. He estado mirando por encima los ejemplos Turbo C, y creo que se pueden escribir en C programas de gestión que se puedan compilar directamente tanto en C68 y Turbo C sin hacer ninguna modificación. Esto último es la ley del mínimo esfuerzo y lo único que puede abaratar costes y tiempo en la elaboración de una aplicación. Eso sí, la transportabilidad tiene como contra no poder utilizar las facilidades extra que ofrece una máquina en particular.

Hoy en día me atrevería a decir sin riesgo a equivocarme que existen demasiados libros para aprender C traducidos al castellano (hay de todo, desde gigantes de 1.000 páginas a libros de bolsillo de 100 páginas). Eso sí, para aprender C solamente necesitáis el libro que he recomendado, el compilador C68, y os sobra.

Salvador MERINO, 1992 07 13

Practicando con el C68 (I)

¿Salvador escribiendo en lenguaje C en vez de en Forth? ¡No!, no están viendo visiones, es todo real.

Si bien conocí el Forth antes que el C, me sentí atraído también por el C años más tarde, aunque siempre me ha gustado más la sencillez y filosofía del Forth. Pero el mayor problema que tenía con el C, dejando atrás todos los inconvenientes de un compilador (el Forth está a caballo entre intérprete y compilador), es que hasta hoy, y a pesar de que disponía de todos los compiladores de C para QL, no me había decidido nunca por cuál compilador utilizar, pues todos tenían sus pros y sus contras. Ayer recibí la versión 3.01 del C68 y la librería QPTR de QJUMP, y creo que por fin tengo el compilador de C estándar para QL (especialmente porque es de dominio público, pero también porque es -sin lugar a dudas- el mejor).

Este es el primer artículo de un montón, pues se supone que voy a traducir toda la librería de subrutinas en BASIC Microsoft, que he aprendido en el curso de analista programador, al C68.

He de advertir que no voy a intentar enseñar a programar en C (¿Qué más quisiera yo?), ya que mis conocimentos de C se reducen a un par de libros que compré hace ya muchos años (espero encontrar un libro con las definiciones ANSI estándar, porque en el manual del C68 solamente viene la lista de funciones ANSI, Unix, Lattice, disponibles), simplemente voy a poner a vuestra disposición una colección de subrutinas o funciones, las cuales os permitirán escribir programas de gestión en un santiamén.

Las funciones que debéis observar en el listado del fichero linput_c con detenimiento son LINPUT y TRIM.

La función LINPUT(cadena, longitud, columna, fila) es un editor de campo, y es muy similar al INPUT del SuperBASIC, pero tiene las siguientes ventajas:

- LLeva incluido en sus parámetros la columna y fila donde se edita la cadena en pantalla.
- Si la cadena es mayor que la longitud dada, es reducida al tamaño de dicha longitud. Y si es menor, se llena de espacios hasta el tamaño dado.
- Devuelve el valor ASCII que hizo terminar la entrada del campo. Esto es muy interesante, pues ese valor nos va a servir para escribir más tarde un editor de formularios o de listas.
- Si se produce un error (pasarse de los límites), tendremos una advertencia mediante un pequeño pitido.

La versión definitiva o actualizada de LINPUT obtendrá los parámetros "columna" y "fila" desde variables globales y tendrá una pequeña modificación que consistirá principalmente en comprobar una variable global, que según su valor, toda entrada sería en minúscula, mayúscula o ambas a la vez. La función TRIM(cadena) es muy maja, sirve para eliminar todos los espacios al comienzo, al final y todos menos uno entre palabras. La función AT(columna, fila) posiciona el cursor en la salida a pantalla por defecto en la emulación Unix. La función CLS() borra la pantalla por defecto. No la utiliza el presente programa, pero queda bonita.

Junto a este artículo, os envío el código fuente y compilado para que podáis probar el programa, el cual lo único que hace es:
- Editar un campo.
- Mostrar el campo completo y el código de finalización.
- Mostrar el campo después de un TRIM.
El código para finalizar el programa es ESC.

He de advertir que el presente programa en C ha sido escrito y compilado con C68 v2.00, pues la v3.01 tiene una rutina de inicialización de la pantalla diferente. También debo advertir que programas escritos en C68 v1.XX es muy probable que no se puedan compilar correctamente con C68 v2.XX, pues hay muchas funciones que se le han cambiado el orden de los parámetros. Lo mismo digo para el C68 v3.XX, programas escritos en C68 v1.XX o V2.XX hay que modificarlos para poderlos compilar correctamente. Asi que como podeis observar, el tema de la transportabilidad es un quebradero de cabeza incluso usando el mismo compilador de C.

Este pequeño programita podría haberse escrito directamente en SuperFORTH sin ninguna dificultad, pero creo que escribir todo el material aprendido en el cursillo de analista programador en C68 tendrá muchísima más aceptación por todos vosotros, pues solamente hay que observar la lista de socios que han pedido el compilador C68 directamente a UK o a la librería de QLíper.

Para compilar el presente programa con el C68 v3.01 hay que cambiar la línea 2 (char _conname[]="con_512x256a0x0";) por:

```
#include <qdos.h>
struct WINDOWDEF _condetails = {2,1,0,7,512,256,0,0};
```

Esto inicia la pantalla stdin, stdout, stderr lo más grande posible y con un borde.

Para saber en qué consistía el nuevo cambio de inicialización de la consola, he tenido que pasar por impresora los ficheros QDOSC68_DOC, LIBC_DOC, LIBQDOS_DOC y LIBM_DOC, y a pesar de todo, me la he visto un poco negras, pues pensaba que el fichero stdio.h llamaba a otros, pero no es verdad en esta versión.

Si de verdad deseáis aprender C, no sería mala idea probar y ver qué pasa, pero tampoco lo sería buscar libros de C que expliquen lo que en el manual del C68 se da por tan conocido e incluso obsoleto en el mundo Unix que ni siquiera se hace un esfuerzo por describir qué hacen.

El mejor libro de C que tengo es "Programación en C. Introducción y conceptos avanzados" (Anaya Multimedia). Fue el primer libro sobre C traducido al castellano. Tiene 496 páginas. En 1988 me costo 4.770 ptas. Os puedo garantizar que el libro es muy bueno.

Salvador Merino, 1992 07 03/06

C68 v3.01

Esta nueva versión tiene como mayor novedad soportar punto flotante IEEE. Las diferencias desde la última versión disponible en QLíper (la v2.00 de enero) son:

(Versión oficial en Inglés)

Release 2.01 (available 8th March, 1992)

Changes

1. The new C68_MENU v4.0 menu-driven front-end and associated documentation and source added to the C68 system. This is the first full release of C68_MENU, although a simple prototype of this program was distributed with Release 2.00 (for evaluation and feedback purposes).

2. Support for Environment Variables added to C68. Various library routines upgraded to achieve this. Also routine supplied for setting/changing environment variables from SuperBasic. Documentation updated appropriately.

3. The new routines fusefd(), fusechid() and usechid() added to the LIBC_A library.

4. Dummy routines for signal handling added to LIBC_A, and signal.h header file added.

5. Assembler versions of many of the string handling routines added to LIBC_A in place of the (slower) C ones.

6. Further small upgrades made to the tutorial to reflect feedback as a result of the fact that flaoting point now works correctly.

7. First version of the LIBQPTR library plus associated documentation and examples made available. This is an extra disk that is only distributed when specifically requested. The current version needs further work so is only sutiable for experienced users of PE.

8. INDENT utility added to distribution (intial QL port contributed by Johnathan Hudson).

9. The routines ANSI routines qsort() and bsearch() added to LIBC_A. Similar LATTICE ones dqsort(), fqsort(), lqsort(), sqsort() and tqsort() also added.

10. UNPROTO utility upgraded to Usenet Release 1.5.

11. CC upgraded to use Environment variables TEMP, TMP and C68PATH.

Distribution Problems/Changes

1. The documentation file UUENCODE_DOC added as it had been omitted.

2. The file MAKE_H was missing from the SOURCE 2 disk, and has now been added.

3. The file CMDS_SED_secomp_c on the SOURCE file had got corrupted - this has now been fixed.

4. The source for ARC, UUD and UUE has been removed from SOURCE 5 (which had filled up) and have now been moved to a separate ARCHIVERS disk.

5. Some of the less common libraries moved to the UTILITIES disk to make room for C68_MENU and associated utilities on

Bugs Fixed

1. Problem with lsbrk() routine in LIBC_A that would cause C68 programs to crash under SMS2 (no affect under QDOS).

2. Problem with _getcdb() routine in LIBC_A. Meant routines using this such as fstat() and stat() did not work correctly.

3. stdout was being opened in "Line Buffered" mode. It is now opened in "unbuffered" mode if using a console.

4. stdin did not have a cursor activated in it by default under certain conditions. A cursor is now explicitly activated as long as it is a console channel.

5. A bug in the strtol() library routine in LIBC_A caused it to often return 0. This also affected routines which called it such as atoi() and atol().

6. A bug in the strtoul() library routine in LIBC_A could cause it to return the wrong value for certain string values.

7. The function version of tolower() in LIBC_A was incorrect. The macro version in ctype_h was OK.

8. The rint() routine in LIBM_A had its name mispelt!

9. A bug in C68 which could cause the compiler to generate a general failure message if you tried to return a value within a void function. You will also now get a warning if you do not return a value when a function is not void!

Release 3.00 (available 30th May, 1992)

Changes

1. Various upgrades done to the C68 compiler itself:
a) IEEE floating point support. This involves changes to the compiler and new support routines.
b) Enhancements to the handling of errors and warning messages. This includes improving the quality of the existing messages, and adding a number of new ones.
c) Improvements to the code optimiser.
d) ANSI style function definitions and prototypes accepted.
e) Parameter type checking added.

2. All library routines that used floating point arithmetic re-compiled to use the IEEE standard.

3. C68menu upgraded to v4.1. Fixes problem with makefile that does not already exist being flagged as 'Alien format'.

4. The following new routines added to LIBC_A library:

```
        strtod()
        atof()
        bcmp()         }
        bcopy()        }
        bzero()         }  for BSD Unix compatibility
        index()        }
        rindex()       }
```

5. The following new header files added to the distribution:
memory_h
sys_times_h

6. The LIBMALLOC_A library upgraded to patchlevel 7 (from usenet). This provides significant changes to the functionality. Both source and documentation changed.

7. The AS68 assembler upgraded to v2.0. This is a port of the Sozobon Assembler v2.0 modifed to produce SROFF output.

8. The SuperBasic part of the Environment Variables support routines enhanced with the addition of the new keywords GETENV$, ENV_LIST and ENV_DEL.

Distribution Problems/Changes

1. The INTRO_DOC file on RUNTIME3 renamed to QDOSC68_DOC as being a more appropriate name.

2. The IOS1_H header file removed from the RUNTIME 1 disk as the definitions it contains are all internal to the Standard C library, and are now in the LIBC_H file provided with the library source.

Bugs Fixed

1. LIBC_A routines 'ut_con' and 'ut_src' were not set up to be globally visible.

2. The order of special options in the command line was significant due to a bug in the scanning routine.

3. The SuperBasic environment variable support module ENV_BIN did not work under Minerva and had problems with error handling.

4. The new version of AS68 fixes the following bugs that occured in the previous version:
a) PC relative instructions were assembled with an incorrect displacement.
b) BTST instructions were not assembled correctly.

Release 3.01 (available 14th June, 1992)

Changes

1. The C68 program enahnced to add further ANSI'fication features and correct minor problems found with the new prototype handling capability.

2. The options added to control the startup process more exactly:
a) Omit code for parsing parameters if not required.
b) Omit code for taking channels from SuperBasic if not required.
c) Allow wildcards in arguments to be expanded automatically.
d) Ability to specify the look of the stdout() channel more exactly.

3. Many of the standard programs relinked to get the new "Title Bar" style start-up windows.

4. The following routines added to the maths library (thanks to Eggert Petersen):

acos	acosh	asinh	atan	atanh
cosh	gamma	lgamma	sinh	tanh

5. Revised/improved versions of the following routines added to the maths library (thanks to Eggert Petersen):

atana	atan2	cos	log	logf
log10	log10f	sin	tan	

6. Documentation for maths library largely re-written.

7. The CPROTO utility added to the RUNTIME2 disk.

8. The following routines added to standard C library LIBC_A.
 stpblk (for Phil Borman!)

Distribution Problems/Changes

1. An obsolete BOOT file was include on the RUNTIME 1 disk. The correct one was substituted.

Bugs Fixed

1. Bug in readdir() routine corrected. A positioning statement was wrong which caused an infinite loop!

KNOWN BUGS

The following is a list of bugs/problems known to be present in the current release of C68.

1. The documentation for the Standard C library (LIBC_A) refers to a number of routines that are not present in this distribution . The routines affected are:
 stc... family of string routines
 stp... family of string routines

2. The documentation for the Maths library (LIBM_A) refers to a number of routines that are not (yet) present in the C68 distribution. Any contributions for missing routines will be gratefully received.

PROBLEMS REPORTED

The following are problems that have been reported as being present, but which have not yet been investigated any further. 1. the 'read_dir' library routine is treating file type 2 as if it were a directory!.

2. The ARC utility accepts a maximum of 13 characters in a filename.

3. If both stdin and stdout are re-directed from the command line, then no stderr channel is opened to the console (even though it was not re-directed).

Developments being worked for future releases

The following is a list of various enhancements that are known to be in hand that will be incorporated in future releases of C68 as they get completed.

1. The development of a semaphore library (Jeremy Allison)

2. A port of the ELVIS editor. Elvis is a Public Domain VI clone for those who know Unix editors. (Dave Walker ?).

3. A port of the RCS v5.6 source control system. (Dave Walker).

4. Ports of an AWK, YACC and FLEX unix style utility programs.

5. Support for Run-time Dynamic Link Libraries. This will allow a set of library routines to be loaded at BOOT time and shared by all C68 programs. (Lester Wareham and Dave Walker)

6. A library (LIBTHING_A) to provide the support for the THING system. (Eric Slagter and Lester Wareham)

La pantalla del QL

Es posible que este tema sea sobradamente conocido por todos, y que por lo tanto carezca de interés. De todas formas me voy a permitir escribir algunas líneas, aunque sólmente sea para refrescar algunos datos y conceptos que hayan podido quedar ligeramente olvidados.

Aunque en nuestro QL disponemos de dos modos de pantalla diferentes, modo 4 (de alta resolución) y modo 8 (de baja resolución), me voy a referir exclusivamente a la alta resolución, ya que el paso a la baja resolución, resultará muy fácil partiendo de la información que voy a facilitar.

1.- Representación de caracteres en modo 4.

Veamos, para empezar,las matemáticas de la memoria de la pantalla, en modo 4 ó 512 (modo de alta resolución).

1.1.- MEMORIA DE PANTALLA

Comienzo área de memoria de pantalla........... 131.072 octetos
Fin área de memoria de pantalla................. 196.607
Total memoria de pantalla....................... 32.768

Distribución de la memoriade pantalla:
Cada línea horizontal ocupa..................... 64 octetos
Hay un total de 512 líneas horizontales
Por lo tanto tenemos: 512 * 64 octetos............ 32.768

1.2.- PIXELES (PUNTOS LUMINOSOS)

Cada línea horizontal tiene 64 octetos * 8 bits... 512 pixeles
Cómo hay 512 líneas horizontales * 512 pixeles... 262.144

Comoquiera que cada dos líneas horizontales forman una sóla línea de puntos luminosos representables o "dibujables", tenemos: 262.144 pixeles / 2 = 131.072

Puntos luminosos representables................. 131.072

Esto se debe, como todos sabemos, a que cada dos octetos, uno par y otro impar, forman un total de 8 puntos luminosos representables. Es decir, cada línea horizontal representa 512 puntos luminosos y hay un total de 256 líneas horizontales representables.

512 * 256....................................... 131.072

1.3.- CARACTERES POR PANTALLA

Los caracteres en pantalla se forman por una matriz de 5 * 9 puntos luminosos, dentro de un rectángulo de 6 * 10. Esto significa que los caracteres no se tocan nunca. Por esto cada caracter se determina por 10 octetos, y de cada uno de esos 10 octetos toma sólamente 6 bits.

512 puntos luminosos / 6 puntos por carácter... 84 caracteres por línea
256 puntos luminosos / 10 puntos por carácter.. 25 caracteres por columna

1.4.- VENTANA POR OMISION

La ventana por omisión es de 448 x 448 a 32 x 16. En esta ventana se pueden representar los siguientes caracteres:

448 puntos luminosos / 6 puntos por carácter.... 74 caracteres por línea
180 puntos luminosos / 10 puntos por carácter... 18 caracteres por columna

El primer octeto de esta ventana de omisión, para el primer carácter superior izquierdo, está en las localizaciones de memoria 133.128/133.129.

2.- Ejemplo práctico de representación de caracteres.

Voy a detallar cómo serían representados los caracteres A B en la parte iquierda de la primera línea de pantalla, y los caracteres C D en la parte izquierda de la segunda línea de pantalla (ventana por omisión).

2.1.- REPRESENTACION

```
    ---------------------------------------------------------------------------
    Memoria  Valor        B i t s          !     B i t s                Valor  Memoria
    octetos    DEC  7  6  5  4  3  2  1  0 !7  6  5  4  3  2  1  0  DEC  octetos
    ----------------------------------------------------------------------------
1   133.128   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  133.130
    133.129   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  133.131

    133.256   57 !  !  !X !X !X !  !  !X !X !X !X !  !  !  !  !  ! 224  133.258
    133.257   57 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  ! 224  133.259

    133.384   69 !  !X !  !  !  !X !  !X !  !  !  !X !  !  !  !  !  16  133.386
    133.385   69 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  16  133.387

    133.512   69 !  !X !  !  !  !X !  !X !  !  !  !X !  !  !  !  !  16  133.514
    133.513   69 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  16  133.515

    133.640  125 !  !X !X !X !X !X !  !X !X !X !X !  !  !  !  !  ! 224  133.642
    133.641  125 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  ! 224  133.643

    133.768   69 !  !X !  !  !  !X !  !X !  !  !  !X !  !  !  !  !  16  133.770
    133.769   69 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  16  133.771

    133.896   69 !  !X !  !  !  !X !  !X !  !  !  !X !  !  !  !  !  16  133.898
    133.897   69 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  16  133.899

    134.024   69 !  !X !  !  !  !X !  !X !X !X !X !  !  !  !  !  ! 224  133.026
    134.025   69 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  ! 224  133.027

    134.152   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  133.154
    134.153   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  134.155

    134.280   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  134.282
    134.281   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  134.283
    ---------------------------------------------------------------------------
```

Para que la representación de los caracteres C D coincida con una nueva página, voy a dejar en blanco las restantes líneas de esta página. Así se podrán observar con una mayor facilidad.

```
    ---------------------------------------------------------------------------
    134.408   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  134.410
    134.409   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  134.411

    134.536   57 !  !  !X !X !X !  !  !X !X !X !X !  !  !  !  !  ! 224  134.538
    134.537   57 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  ! 224  134.539

    134.664   69 !  !X !  !  !  !X !  !X !  !  !  !X !  !  !  !  !  16  134.666
    134.665   69 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  16  134.667

    134.792   65 !  !X !  !  !  !  !  !X !  !  !  !X !  !  !  !  !  16  134.794
    134.793   65 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  16  134.795

    134.920   65 !  !X !  !  !  !  !  !X !  !  !  !X !  !  !  !  !  16  134.922
    134.921   65 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  16  134.923


    135.048   65 !  !X !  !  !  !  !  !X !  !  !  !X !  !  !  !  !  16  135.050
    135.049   65 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  16  135.051

    135.176   69 !  !X !  !  !  !X !  !X !  !  !  !X !  !  !  !  !  16  135.178
    135.177   69 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  16  135.179

    135.304   57 !  !  !X !X !X !  !  !X !X !X !X !  !  !  !  !  ! 224  135.306
    135.305   57 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  ! 224  135.307

    135.432   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  135.434
    135.433   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  135.435

    135.560   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  135.562
```

```
    135.561   00 !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  !  00  135.563
    ---------------------------------------------------------------------------
```

El carácter X representa un punto luminoso. He considerado conveniente representar estos cuatro caracteres formando un "conjunto", para que se aprecie mejor cómo se combinan sus "bits informativos" en los correspondientes octetos.

2.2.- OCTETOS PAR E IMPAR

En el modo de alta resolución, al que estamos haciendo referencia, cada dos octetos contiguos, una palabra sencilla, definen ocho puntos luminosos. El primer bit de definición se encuentra en el primero de los octetos, el octeto par, (bit número 7) y el segundo bit de definición se encuentra en el segundo de los octetos, el octeto impar, (bit número 7). Igualmente el tercer bit de definición se encuentra en el octeto par, (bit número 6). y el cuarto bit de definición se encuentra en el octeto impar, (bit número 6). Y así sucesivamente hasta completar los 8 bits de ambos octetos, par e impar, para continuar en las siguientes parejas de octetos. Cada pareja de "bits de definición" determina un "punto luminoso".

2.3.- COLOR DE LOS CARACTERES

Lo octetos pares contienen el "bit de definición" del "verde" de un punto determinado, y los octetos impares contienen el "bit de definición" del" rojo" de este punto. De esta forma, si ninguno de los dos bits está activado (0), el punto luminoso será negro. Si ambos bits están activados (1), el punto luminoso será blanco. Si sólamente el bit del verde está activado (1), el punto luminoso será verde. Y si finalmente sólamente el bit del rojo está activado (1), el punto luminosos será rojo.

En el ejemplo se ha considerado que todas las parejas de octetos tienen el mismo valor, por lo que los caracteres aparacerán en blanco, sobre fondo negro.

3.- Posiciones de memoria.

A continuación voy a indicar unos sencillos cálculos que nos permitirán conocer las localizaciones de memoria que contienen los octetos con la información de un determinado punto luminoso en la pantalla.

3.1.- COORDENADAS DEL PUNTO LUMINOSO Y FORMULAS

Si llamamos "X" e "Y" a las coordenadas horizontal y vertical respectivamente de un punto real de la pantalla, contando desde la parte superior izquierda de la misma, (0,0), tenemos:

```
    Localización del octeto par ( con el bit de definición del verde)
                   131.072 + Y * 128 + INT ( X / 8 ) * 2
    Localización del octeto impar ( con el bit de definición del rojo )
                   131.072 + Y * 128 + INT ( X / 8 ) * 2 + 1
    Posición del "bit de definición" dentro del octeto
                   7 - ( X - INT ( X / 8 ) * 8 )
```

Con estos sencillos cálculos podemos conocer las posiciones de memoria de los correspondientes octetos, y nos permitirá poder "pokear" o "peekear" para escribir o leer la información deseada.

Félix Alonso, 1992 05

El malabarista del Amiga

¿Quién lo iba a decir? ¡La famosa demo del malabarista del Commodore Amiga pasada al QL! Lo que es más gracioso, del formato original de esta animación creada en el Amiga mediante algún programa de "ray tracing" en 4096 colores se ha pasado al formato de baja resolución del QL (8 colores), pasando previamente por una conversión del Amiga al Atari ST en baja resolución en 16 colores. Y aun así resulta curioso verlo en acción.

Para poder incluir en un disco estándar de formato QL el mayor número de

pantallas de animación se han utilizado unos procedimientos de compresión y expansión de pantallas sacadas de la revista QL World de diciembre de 1986, cuyo autor es James Lucy. Para obtener las pantallas en formato QL se ha utilizado un programa para pasar formatos ST a QL, cuyos autores son Alan Pemberton y Rich Mellor.

Mi único merito ha sido tener la paciencia de ir pasando una a una las pantallas, que a su vez obtuve mediante un programa de dominio público para el Atari ST a partir de una animación, donde todas las pantallas están integradas en un solo fichero.

Para lograr un resultado decente es necesario tener un sistema formado por un Atari más tarjeta emuladora o Commodore Amiga más emulador, o mejor aún, un QL equipado con una Gold Card. En cualquiera de los casos conviene tener abundante memoria RAM disponible, para poder utilizar un disco RAM, en cuyo caso se obtendrá una animación bastante fluida. Si sólo se dispone de un QL básico con unidad de disco, entonces la animación será un poco lenta...

Este programa es del dominio público [Red.: estará disponible en la librería de QLíper, ya que por su extesión es imposible incluirlo aquí]. Su única intención es la de entretener y ofrecer una curiosidad a la que los usuarios del QL no solemos estar muy acostumbrados. Si alguna de las personas a las que llegue este disco quiere hacerme algún comentario sobre el mismo o con cualquier tema relacionado con el fascinante mundo del QL podéis contactar conmigo en la dirección que a continuación paso a detallar.

Javier ZUBIETA
Novia Salcedo 26 - 4º izda.
E-48012 BILBAO

Las cuentas de QLíper, y la cuota de 1993

En el número 41 aparecerán por última vez los movimientos de caja de 1992, y empezaremos el nuevo año a partir del saldo de entonces, así que por penúltima vez podéis echar un vistazo a la lista completa y actualizada:

Fecha	Concepto	Ptas.	Saldo
91.10.11	Suscripción de Dasio CARBALLEIRA	+1500	+1500
91.10.14	Suscripción de Félix GALLEGO	+1500	+3000
91.10.15	25 sobres de papel reciclado	-75	+2925
91.10.17	Suscripción de Javier ZUBIETA	+1500	+4425
91.10.17	Sellos (5 de 25 y 5 de 15) para cartas	-200	+4225
91.10.22	Suscripción de Salvador MERINO	+1500	+5725
91.10.23	50 discos de 3.5 pulgadas sin marca	-3000	+2725
91.10.30	Suscripción de Luis VALERO	+1500	+4225
91.11.18	Suscripción de Felipe BERGANZA	+1500	+5725
91.12.10	Suscripción de Julián COLOMINA	+1600	+7325
91.12.11	50 sobres acolchados	-1900	+5425
91.12.12	Suscripción de Josu REGIDOU	+1500	+6925
91.12.14	Suscripción de Alfredo GRACIA	+1500	+8425
91.12.14	Suscripción de Pedro REINA	+1500	+9925
91.12.14	Suscripción de Félix ALONSO	+1500	+9825
91.12.16	Sellos (10 de 25) para cartas	-250	+11425
91.12.20	Suscripción de Mariano BERGES	+1500	+12675
91.12.20	Sello de 45 para carta	-45	+12630
91.12.20	40 sobres de papel reciclado	-120	+12510
91.12.27	Suscripción de Francisco DIAZ-TENDERO	+1500	+14010
91.12.27	Suscripción de Pablo CARDENES	+1500	+15510
91.12.27	400 etiquetas en papel continuo	-2635	+12875
92.01.04	Suscripción de Miguel ESTARELLAS	+1500	+14375
92.01.04	Suscripción de José Carlos DE PRADA	+1500	+15875
92.01.04	Suscripción de Pablo POL	+1500	+17375
92.01.04	Suscripción de Fco. Javier OLIVAN	+1500	+18875
92.01.07	Sellos para el QLíper 36	-645	+18230
92.01.17	Suscripción de Rafael ILLANES	+1500	+19730
92.01.18	Sello de 6 pesetas para el QLíper 36	-6	+19724
92.03.03	Sellos (6 de 26) para el QLíper 37	-156	+19568

92.03.03	50 discos de 3.5 sin marca	-3000	+16568
92.03.04	Sellos (12 de 36) para el QLíper 37	-432	+16136
92.03.04	Sello de 57 para el QLíper 37	-57	+16079
92.04.28	Sellos (2*57+17*36) para el QLíper 38	-726	+15353
92.06.29	1330 etiquetas adhesivas en continuo	-1505	+13848
92.06.29	100 sobres para enviar los discos	-400	+13448
92.06.30	Sellos (3*57+16*37) para el QLíper 39	-753	+12695
92.08.10	Sellos (2*27)	-54	+12641
92.08.18	Sellos para envío aéreo a Chile del nº39	-525	+12116
92.08.24	50 discos sin marca	-3250	+8866
92.09.09	Sellos (¡uf!) para QLíper 40	-2690	+6176
92.09.14	Sellos para circular sobre QLíper 40	-1076	+5100
92.10.13	Suscripción de Dasio CARBALLEIRA	+2000	+7100
92.10.13	Contribución de Dasio CARBALLEIRA	+1000	+8100
92.11.17	10 sobres acolchados medianos	-360	+7740
92.11.17	40 sobres acolchados pequeños	-1120	+6620
92.11.17	50 discos sin marca	-3250	+3370
Total actual en caja:			+3370

Los discos de QLíper:

Fecha	Cantidad	Total
91.10.23	+50	+50
92.01.07	-20 (QLíper 36)	+30
92.03.03	+50	+80
92.03.04	-20 (QLíper 37)	+60
92.04.25	-20 (QLíper 38)	+40
92.06.30	-20 (QLíper 39)	+20
92.08.24	+50	+70
92.09.09	-34 (QLíper 40)	+36
92.11.17	+50	+86
92.11.18	-34 (QLíper 41)	+52
	-05 (Redacción)	+47
		=====
Discos disponibles para enviar: ..		+47

LA CUOTA DE 1993

Decidir la cuota no ha sido fácil, porque ha habido que jugar con tres parámetros:

1) El precio de los sobres. Aún no está claro qué sobres usaremos normalmente. Lo ideal sería usar un sobre rígido de cartón para discos, pero no he sido capaz de localizarlo, aunque sé que existen, al menos en otros países (recibí uno ideal de EE.UU., del Forth Interest Group). Los sobres "de nóminas" (así se llaman, según me dijeron en la tienda) que hemos usado últimamente, son muy baratos, pero hay que usar un cartón para proteger el disco, y tanto cartoncito (que hay que buscar, recortar, pegar, meter...) es de locos, ya os lo podéis imaginar. Por tanto, mientras encontramos la solución ideal, probablemente tengamos que olvidar nuestros principios por cierto tiempo cual miserables ratas mercenarias, y usar sobres acolchados. Por tanto, para no pillarnos los dedos, tenemos que considerar que usaremos esos sobres a lo largo del año. El riesgo es que no sabemos el precio de los sobres de cartón que andamos buscando, pero es muy improbable que sean más caros que los acolchados.

2) El número de suscriptores. Somos 20 (19 pagáis oficialmente, y uno curra y paga los gastos extraordinarios). Es de temer que la "población" disminuya en 1993, por múltiples causas de todos imaginables. Por otro lado, el gasto ha aumentado con el intercambio a los grupos extranjeros, que de momento es sólo una experiencia. Si el número de suscriptores se reduce considerablemente, nos podemos ir despidiendo de esto. Sabéis que hay mucha gente (+-40) en la base de datos, y que muchas veces he mostrado mi intención de enviarles una circular, y que aún no lo he hecho. El motivo principal es que apenas podemos ofrecerles nada. Como Salvador me dijo, sería un derroche de dinero para un grupo tan pequeño como nosotros. Pero yo sigo pensando que hay que hacerlo, porque siempre hay posibilidad de encontrar a algún Robinsón, y al menos así limpiaremos de "conversos" la base de datos. Pero evidentemente no se puede hacer de cualquier manera, hay que preparar una carta en condiciones, en la que se muestre qué hacemos, en la que se hable de los nuevos proyectos y la difusión del QDOS

(emuladores, compatibles, etc.). Todo ello con WordPerfect (¡traición!) e impresora láser. ¿No os parece? Y convendría hacerlo ya, así que nos vamos a poner manos a la obra.

3) El número de QLíperes al año. He barajado la posibilidad de que sacásemos sólo 4 números, coincidiendo con las estaciones (qué poético), y así mantener la cuota actual, pero era un cambio muy fuerte. Creo que el ritmo actual es correcto de momento.

Al final, el resultado ha sido, consultado con Salvador Merino y Pedro Reina, el siguiente: 6 números al año y cuota de 2000 pesetas.

En fin, todas estas elucubraciones han sido realizadas gracias a una hoja de Abacus diseñada para tal fin. Para que podáis jugar con ella, se encuentra en este disco con el nombre de pesetas_aba. Probad a cambiar el precio de los sobres (4 pesetas los de ahora, 28 ó 36 los acolchados), el número de QLíperes al año (4 ó 6), el número de suscriptores, la cuota (1500 ó 2000), y veréis cuál es el saldo final que resulta para un año. ¡Que os divirtáis!

Por último, deciros que los métodos para pagar la cuota son:

1) Oculta en sobre bien cerrado. El más cómodo y barato (y el más arriesgado, si no va certificado).
2) Mediante giro postal (el más seguro).
3) Talón. El más incómodo (para mí) y el más caro (para vosotros). Incómodo porque mi carácter noctámbulo (y alevoso) me pone casi siempre muy difícil ir por la mañana a ningún sitio (bancos incluidos), aunque también puede meterse en un cajero. Caro porque los ladrones de cuello blanco del Banco Santander cobran 300 pesetas por cheque tramitado, con lo cual os agradecería que lo extendieseis por 2300 pesetas.

Si fuésemos un grupo algo más numeroso, creo que merecería la pena tener una cuenta en la Caja Postal (por la facilidad de acceso desde cualquier oficina de Correos), cuyos extractos serían enviados regularmente a los suscriptores. ¿No os parece? Cualquier propuesta o idea sobre esto, como sobre cualquier otro aspecto de QLíper, es bienvenida.